

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ

Кафедра системного програмування і спеціалізованих комп'ютерних систем

До захисту допущено

Завідувач кафедри

_____ **Віталій РОМАНКЕВИЧ**
(підпис) (ініціали, прізвище)

“ ____ ” _____ 202__ р.

Дипломний проєкт

на здобуття ступеня бакалавра

за освітньо-професійною програмою «Системне програмування»

спеціальності

123 «Комп'ютерна інженерія»

на тему: Програмні засоби автоматизованого обліку та аналізу даних успішності студентів ЗВО

Виконав:

студент IV курсу, групи КВ-61

Керімов Хікмет Немат огли

(прізвище, ім'я, по батькові)

_____ (підпис)

Керівник доц. каф. СПіСКС, к. т. н., доцент Петрашенко А.В.

(посада, науковий ступінь, вчене звання, прізвище та ініціали)

_____ (підпис)

Консультант з нормоконтролю, доц.каф.СПСКС, к.т.н. Клятченко Я.М.

(назва розділу) (посада, вчене звання, науковий ступінь, прізвище, ініціали)

_____ (підпис)

Рецензент _____

(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище та ініціали)

_____ (підпис)

Засвідчую, що у цьому дипломному
проєкті немає запозичень з праць інших
авторів без відповідних посилань.

Студент _____
(підпис)

Київ – 2020 року

ВІДОМІСТЬ ДИПЛОМНОГО ПРОЄКТУ

| № з/п | Формат | Позначення | Найменування | Кількість листів | Примітка |
|-------|--------|--------------------|------------------------------|------------------|----------|
| 1 | A4 | | Завдання на дипломний проєкт | 2 | |
| 2 | A4 | ІАЛЦ.045440.001 ОА | Опис альбому | 2 | |
| 3 | A4 | ІАЛЦ.045440.002 ТЗ | Технічне завдання | 4 | |
| 4 | A4 | ІАЛЦ.045440.003 ТП | Відомість технічного проєкту | 2 | |
| 5 | A4 | ІАЛЦ.045440.004 ПЗ | Пояснювальна записка | 50 | |
| 6 | A4 | ІАЛЦ.045440.005 Д1 | Схема структурна | 1 | |
| 7 | A4 | ІАЛЦ.045440.006 Д2 | Схема структурна | 1 | |
| 8 | A4 | ІАЛЦ.045440.007 Д3 | Схема алгоритму | 1 | |
| 9 | A4 | ІАЛЦ.045440.008 Д4 | Схема алгоритму | 1 | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |

| | | | | | | |
|-----------|------------|-------|------|------------------------------|--|--------|
| | | | | ДП ІАЛЦ.045440.000 | | |
| | ПІБ | Підп. | Дата | | | |
| Розробн. | Керімов | | | Відомість дипломного проєкту | Лист | Листів |
| Керівн. | Петрашенко | | | | 1 | 1 |
| Консульт. | Клятченко | | | | КП ім. Ігоря Сікорського Каф. СПіСКС Гр. КВ-61 | |
| Н/контр. | Клятченко | | | | | |
| Зав.каф. | Романкевич | | | | | |

Пояснювальна записка до дипломного проєкту

на тему: Програмні засоби автоматизованого обліку та аналізу даних успішності
студентів ЗВО

Київ – 2020 року

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ

Кафедра системного програмування і спеціалізованих комп'ютерних систем

Рівень вищої освіти – перший (бакалаврський)

Спеціальність 123 «Комп'ютерна інженерія»

Освітньо-професійна програма «Системне програмування»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Віталій РОМАНКЕВИЧ

(підпис) (ініціали, прізвище)

«___» _____ 20__ р.

ЗАВДАННЯ

на дипломний проєкт студента

Керімова Хікмета Немат огли

(прізвище, ім'я, по батькові)

1. Тема проєкту "Програмні засоби автоматизованого обліку та аналізу даних успішності студентів ЗВО",

керівник проєкту доц. каф. СПіСКС, к. т. н., доцент Петрашенко А.В.,

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від «25» травня 2020 р. №1181-С

2. Термін подання студентом проєкту _____

3. Вихідні дані до проєкту:

- система автоматизованого обліку та аналізу даних успішності студентів ЗВО у вигляді вебдодатку.

4. Зміст пояснювальної записки:

- аналіз існуючих програмних засобів автоматизованого обліку та обґрунтування теми дипломного проєкту;
- аналіз технологій для розробки програмних засобів автоматизованого обліку та аналізу даних;

- структура проєкту та опис роботи програм;
- тестування проєкту й аналіз отриманих результатів.

5. Перелік графічного матеріалу:

- загальна схема проєкту (схема структурна);
- модель даних (схема структурна);
- основний маршрут користувача (схема алгоритму);
- модуль візуалізації (схема алгоритму).

6. Консультанти розділів проєкту

| Розділ | Прізвище, ініціали та посада консультанта | Підпис, дата | |
|---------------|--|----------------|------------------|
| | | завдання видав | завдання прийняв |
| Нормоконтроль | Клятченко Я.М., к.т.н., доцент каф. СПіСКС | | |

7. Дата видачі завдання “20” грудня 2019 р.

Календарний план

| № з/п | Назва етапів виконання дипломного проєкту | Термін виконання етапів проєкту | Примітка |
|-------|---|---------------------------------|----------|
| 1. | Вивчення літератури за тематикою проєкту | 21.12.2019 | |
| 2. | Розроблення та узгодження технічного завдання | 24.12.2019 | |
| 3. | Аналіз існуючих рішень | 13.01.2020 | |
| 4. | Підготовка матеріалів першого розділу проєкту | 27.01.2020 | |
| 5. | Розробка програмного забезпечення | 17.02.2020 | |
| 6. | Відлагодження програмного продукту | 16.03.2020 | |
| 7. | Підготовка матеріалів другого розділу проєкту | 30.03.2020 | |
| 8. | Розроблення інтерфейсу програмного забезпечення | 27.04.2020 | |
| 9. | Підготовка графічної частини | 04.05.2020 | |
| 10. | Оформлення документації дипломного проєкту | 13.05.2020 | |

Студент

(підпис)

Хікмет КЕРІМОВ

Керівник проєкту

(підпис)

Андрій ПЕТРАШЕНКО

АННОТАЦІЯ

Дипломний проєкт включає пояснювальну записку (50 с., 28 рис., 5 додатків).

Об'єкт розробки – розробка програмних засобів для створення власної системи автоматизованого обліку та аналізу даних успішності студентів закладів вищої освіти (ЗВО).

Перевагою даного проєкту є повністю український інтерфейс користувача, та робота з широко використаними технологіями для реалізації як клієнтської, так і серверної сторони додатку. Результати дипломного проєкту можуть бути застосовані у сфері вищої освіти.

В процесі розробки були використані бібліотеки та фреймворки мови Python, для роботи із серверною частиною вебфреймворк Django, для роботи з даними бібліотека Pandas, та для візуалізації даних бібліотека Matplotlib. В якості системи керування базами даних (СКБД) використовувалась PostgreSQL. В ході розробки:

- проведено аналіз існуючих систем для автоматизованого обліку успішності студентів;
- проведено аналіз існуючих технологій для реалізації даної системи;
- розроблено Веб-додаток для управління роботи системи автоматизованого обліку;
- реалізовано візуалізацію даних у вебдодатку.

Ключові слова: автоматизований облік, аналіз даних, вебдодаток, клієнт, сервер, Django, Pandas, Matplotlib, системи керування базами даних, PostgreSQL.

SUMMARY

Diploma project consists of explanatory note (50 p., 28 img., 5 additions).

The object of development is the creation of software for creating your own system of automated accounting and analysis of student performance data of higher education institutions (HEI).

The advantage of this project is a completely Ukrainian user interface, and work with widely used technologies to implement both the client and server side of the application. The results of the thesis can be applied in the field of higher education.

Python libraries and frameworks were used in the development process, namely:

- to work with the server part Django web framework;
- to work with data Pandas library;
- Matplotlib library for data visualization.

PostgreSQL was used as a database management system (DBMS).

During development:

- the analysis of existing systems for the automated account of success of students is carried out;
- the analysis of existing technologies for realization of the given system is carried out;
- developed a web application for managing the automated accounting system;
- data visualization in the web application is implemented.

Keywords: automated accounting, data analysis, Web application, client, server, Django, Pandas, Matplotlib, database management systems, PostgreSQL.

[illegible]

ЗМІСТ

| | |
|---|---|
| 1. НАЙМЕНУВАННЯ ТА ГАЛУЗЬ РОЗРОБКИ | 2 |
| 2. ПІДСТАВА ДЛЯ РОЗРОБКИ | 2 |
| 3. ЦІЛЬ І ПРИЗНАЧЕННЯ РОБОТИ | 2 |
| 4. ДЖЕРЕЛА РОЗРОБКИ | 2 |
| 5. ТЕХНІЧНІ ВИМОГИ | 3 |
| 5.1. Вимоги до програмного продукту, що розробляється | 3 |
| 5.2. Вимоги до апаратного забезпечення | 3 |
| 5.3. Вимоги до програмного та апаратного забезпечення користувача | 3 |
| 6. ЕТАПИ РОЗРОБКИ | 4 |

| | | | | | | | | | | | | | | |
|------------------|-----------------|-----------------|--------------|-------------|--|--|--|--|--|---------------------------------------|-------------|---------------|---|--|
| | | | | | ІАЛЦ.045440.002 ТЗ | | | | | | | | | |
| | | | | | | | | | | | | | | |
| Зм | Лист | № докум. | Підп. | Дата | | | | | | | | | | |
| Розроб. | Керімов Х.Н.о. | | | | Програмні засоби автоматизованого обліку та аналізу даних успішності студентів ЗВО Технічне завдання | | | | | Лім. | Лист | Листів | | |
| Перев. | Петрашенко А.В. | | | | | | | | | | | 1 | 4 | |
| | | | | | | | | | | КПІ ім. І. Сікорського, ФПМ, КВ-61 | | | | |
| Н. контр. | Клятченко Я.М. | | | | | | | | | | | | | |
| Затв. | Романкевич В.О. | | | | | | | | | | | | | |

1 НАЙМЕНУВАННЯ ТА ГАЛУЗЬ РОЗРОБКИ

Назва розробки: "Програмні засоби автоматизованого обліку та аналізу даних успішності студентів ЗВО".

Галузь застосування: сфера освіти.

2 ПІДСТАВА ДЛЯ РОЗРОБКИ

Підставою для розробки є завдання на дипломне проєктування на здобуття першого (бакалаврського) рівня вищої освіти, затверджене кафедрою системного програмування і спеціалізованих комп'ютерних систем Національного технічного університету України «Київський Політехнічний Інститут імені Ігоря Сікорського».

3 ЦІЛЬ І ПРИЗНАЧЕННЯ РОБОТИ

Метою даного проєкту є створення системи автоматизованого обліку та аналізу даних клієнт-серверної архітектури.

4 ДЖЕРЕЛА РОЗРОБКИ

Джерелом інформації є технічна та науково-технічна література, технічна документація, публікації у періодичних виданнях та електронні статті у мережі Інтернет.

| | | | | | | |
|-----------|-------------|-----------------|--------------|-------------|----------------------------------|-------------|
| | | | | | <i>ІАЛЦ.045440.002 ТЗ</i> | Лист |
| Зм | Лист | № докум. | Підп. | Дата | | 2 |

5 ТЕХНІЧНІ ВИМОГИ

5.1 Вимоги до програмного продукту, що розробляється

- сумісність з операційною системою Linux;
- можливість реєстрації у систему;
- можливість роботи з системою керування базами даних (СКБД) – додавання, зчитування, редагування та видалення даних з неї;
- можливість візуалізації даних про успішність;
- наявність зручного меню.

5.2 Вимоги до апаратного забезпечення

- оперативна пам'ять: 2 Гб;
- наявність доступу або до мережі Ethernet (IEEE 802.3), або до мережі Wi-Fi (IEEE 802.11 b/g/n).

5.3 Вимоги до програмного та апаратного забезпечення користувача

- операційна система Linux;
- наявність браузера;
- наявність доступу або до мережі Ethernet (IEEE 802.3), або до мережі Wi-Fi (IEEE 802.11 b/g/n).

6 ЕТАПИ РОЗРОБКИ

| № з/п | Назва етапів виконання дипломного проєкту | Термін виконання етапів |
|-------|---|-------------------------|
| 1. | Видача завдання на дипломне проєктування | 24.12.2019 |
| 2. | Вивчення літератури за тематикою роботи | 21.12.2019 |
| 3. | Розроблення та узгодження технічного завдання | 24.12.2019 |
| 4. | Розроблення структури додатку | 13.01.2020 |
| 5. | Розроблення дизайну та графічних елементів | 27.01.2020 |
| 6. | Програмна реалізація додатку | 17.02.2020 |
| 7. | Тестування додатку | 16.03.2020 |
| 8. | Підготовка матеріалів текстової частини проєкту | 30.03.2020 |
| 9. | Підготовка матеріалів графічної частини проєкту | 27.04.2020 |
| 10. | Оформлення технічної документації проєкту | 13.05.2020 |

| Поз. | Формат | ПОЗНАЧЕННЯ | НАЙМЕНУВАННЯ | Кількість аркушів | № прим. | Примітки |
|---|-----------------|--------------------|---|-------------------|---------------------------------------|-------------|
| | A4 | IАЛЦ.045440.004 ПЗ | Програмні засоби автоматизованого обліку та аналізу даних успішності студентів ЗВО Пояснювальна записка | 50 | | |
| | A4 | IАЛЦ.045440.005 Д1 | Загальна схема проєкту автоматизованого обліку та аналізу даних успішності студентів ЗВО Схема структурна | 1 | | |
| | A4 | IАЛЦ.045440.006 Д2 | Модель даних системи автоматизованого обліку та аналізу даних успішності студентів ЗВО Схема структурна | 1 | | |
| | A4 | IАЛЦ.045440.007 Д3 | Основний маршрут користувача у системі автоматизованого обліку та аналізу даних успішності студентів ЗВО Схема алгоритму | 1 | | |
| IАЛЦ.045440.003 ТП | | | | | | |
| Зм | Лист | № докум. | Підп. | Дата | | |
| Розроб. | Керімов Х.Н.о. | | | | | |
| Перев. | Петрашенко А.В. | | | | | |
| H. контр. | Клятченко Я.М. | | | | | |
| Завв. | Романкевич В.О. | | | | | |
| Програмні засоби автоматизованого обліку та аналізу даних успішності студентів ЗВО Відомість технічного проєкту | | | | | Лім. | Лист Листів |
| | | | | | | 1 2 |
| | | | | | КПІ ім. І. Сікорського, ФПМ, КВ-61 | |

[illegible]

ЗМІСТ

| | |
|---|----|
| ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ, ТЕРМІНІВ | 3 |
| ВСТУП | 4 |
| 1. АНАЛІЗ ІСНУЮЧИХ ПРОГРАМНИХ ЗАСОБІВ АВТОМАТИЗОВАНОГО ОБЛІКУ ТА ОБҐРУНТУВАННЯ ТЕМИ ДИПЛОМНОГО ПРОЄКТУ | 5 |
| 1.1. Причини використання автоматизованого обліку | 5 |
| 1.2. Огляд принципів роботи програмних засобів автоматизованого обліку | 6 |
| 1.3. Аналіз існуючих програмних засобів автоматизованого обліку | 7 |
| 1.4. Обґрунтування теми дипломного проєкту | 11 |
| Висновки до розділу | 12 |
| 2. АНАЛІЗ ТЕХНОЛОГІЙ ДЛЯ РОЗРОБКИ ПРОГРАМНИХ ЗАСОБІВ АВТОМАТИЗОВАНОГО ОБЛІКУ ТА АНАЛІЗУ ДАНИХ | 13 |
| 2.1. Обґрунтування обраного типу прикладного програмного забезпечення для розробки програмних засобів автоматизованого обліку | 13 |
| 2.2. Порівняльний аналіз програмних платформ для розробки серверних додатків | 18 |
| 2.3. Порівняльний аналіз систем керування базами даних (СКБД) | 29 |
| 2.4. Обґрунтування обраних засобів для аналізу даних | 35 |
| Висновки до розділу | 36 |

| | | | | | | | |
|------------------|-----------------|-----------------|--------------|-------------|---|-------------|---------------|
| | | | | | ІАЛЦ.045440.004 ПЗ | | |
| Зм | Лист | № докум. | Підп. | Дата | | | |
| Розроб. | Керімов Х.Н.о. | | | | Програмні засоби автоматизованого обліку та аналізу даних успішності студентів ЗВО Пояснювальна записка | | |
| Перев. | Петрашенко А.В. | | | | | | |
| | | | | | | | |
| Н. контр. | Клятенко Я.М. | | | | | | |
| Затв. | Романкевич В.О. | | | | | | |
| | | | | | Лім. | Лист | Листів |
| | | | | | | 1 | 50 |
| | | | | | КПІ ім. І. Сікорського, ФПМ, КВ-61 | | |

| | |
|---|----|
| 3. СТРУКТУРА ПРОЄКТУ ТА ОПИС РОБОТИ ПРОГРАМ..... | 38 |
| 3.1. Структура проекту..... | 38 |
| 3.2. Програма для роботи з користувачем "Студент"..... | 40 |
| 3.3. Програма для роботи з користувачем "Викладач"..... | 42 |
| 3.4. Програма для роботи з даними та їх аналізу..... | 42 |
| 3.5. Налаштування СКБД..... | 42 |
| Висновки до розділу..... | 43 |
| 4. ТЕСТУВАННЯ ПРОЄКТУ Й АНАЛІЗ ОТРИМАНИХ РЕЗУЛЬТАТІВ..... | 44 |
| 4.1. Тестування роботи проекту..... | 44 |
| Висновки до розділу..... | 47 |
| ВИСНОВКИ..... | 48 |
| СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ..... | 49 |

ДОДАТКИ

| | | | | | | |
|-----------|-------------|-----------------|--------------|-------------|---------------------------|------|
| | | | | | ІАЛЦ.045440.004 ПЗ | Лист |
| <i>Зм</i> | <i>Лист</i> | <i>№ докум.</i> | <i>Підп.</i> | <i>Дата</i> | | 2 |

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ, ТЕРМІНІВ

ЗВО – заклад вищої освіти.

ОС – операційна система.

ПЗ – програмне забезпечення.

СКБД – система керування базами даних.

API (Application Programming Interface) - інтерфейс програмування додатків.

HTML (Hyper Text Markup Language) - мова розмітки гіпертексту.

HTTP (Hyper Text Transfer Protocol) - протокол передачі гіпертекстових документів.

ORM (Object-Relational Mapping) - об'єктно-реляційне відображення.

P2P (Peer-to-peer) – рівний до рівного.

PHP (Hypertext Preprocessor) - скриптова мова програмування.

SQL (Structured Query Language) - мова структурованих запитів.

URL (Uniform Resource Locator) - єдиний вказівник на ресурс.

| | | | | | | |
|----|------|----------|-------|------|----------------------------------|------|
| | | | | | <i>ІАЛЦ.045440.004 ПЗ</i> | Лист |
| Зм | Лист | № докум. | Підп. | Дата | | 3 |

ВСТУП

Проблеми автоматизованого обліку успішності студентів є однією з найактуальніших у наш час. У минулому цей облік відбувався вручну, викладачі занотовували відвідуваність та успішність студентів. Це є обов'язком викладача, контролювати успішність студентів, адже завдяки цьому можна зрозуміти наскільки ефективним є сам викладач. Якщо студенти показують гарний результат у навчанні, можна зробити висновок, що вони мають досить освіченого та відповідального викладача, який слідкує за навчальним процесом, і завдяки своїй праці демонструє ефективність всього професорсько-викладацького складу. Таким чином, викладачі також показують здатність та уміння готувати кваліфікованих фахівців у певній сфері. В свою чергу, інформація про успішність є важливою для самих студентів. Адже таким чином вони можуть дізнатися, де мають проблеми, і що мають зробити, щоб не відставати від своїх одногрупників. І це розвиває у студентів усвідомлення необхідності навчатися упродовж всього життя, що є досить важливою навичкою, для пристосування у світі сучасних технологій, які змінюються та розвиваються швидкими темпами.

Саме тому, автоматизація такого процесу є досить необхідною, для прискорення та спрощення роботи як викладачів, так і студентів. Враховуючи розвиток сучасних технологій, можливість візуалізувати дані, є неймовірним проривом, і дає можливість викладачам оцінювати студентів більш об'єктивно у порівнянні з минулими роками.

| | | | | | | |
|----|------|----------|-------|------|---------------------------|------|
| | | | | | ІАЛЦ.045440.004 ПЗ | Лист |
| Зм | Лист | № докум. | Підп. | Дата | | 4 |

1 АНАЛІЗ ІСНУЮЧИХ ПРОГРАМНИХ ЗАСОБІВ АВТОМАТИЗОВАНОГО ОБЛІКУ ТА ОБҐРУНТУВАННЯ ТЕМИ ДИПЛОМНОГО ПРОЄКТУ

1.1 Причини використання автоматизованого обліку

Автоматизований облік слугує для покращення і полегшення процесу контролю успішності студентів для викладачів. В свою чергу поняття «облік» являє собою організовану належним чином систему збору, обробки й узагальнення необхідної інформації, в нашому випадку це досягнення студентів, які виражаються балами, і відповідно вони отримують їх наступним чином:

- відвідуванням лекційних, практичних або семінарських занять;
- здачею лабораторних робіт;
- виконанням контрольних робіт;
- здачею екзаменаційної сесії;
- здачею курсових робіт/проєктів;
- здачею дипломних робіт/проєктів.

Вже зараз існує не мала кількість подібних систем, які допомагають мобільно слідкувати за успішністю студентів. Та жодна з них не дозволяє провести аналіз отриманих даних, адже зібравши їх, можна зробити висновки щодо засвоєного матеріалу студентами, відвідування занять, тощо.

1.2 Огляд принципів роботи програмних засобів автоматизованого обліку

Розвиток технологій вплинув на кожную сферу нашого життя, в тому числі й на освіту. Зараз активно відбувається автоматизація ледь не кожного процесу,

| | | | | | | |
|----|------|----------|-------|------|---------------------------|------|
| | | | | | ІАЛЦ.045440.004 ПЗ | Лист |
| Зм | Лист | № докум. | Підп. | Дата | | 5 |

який раніше, у закладах освіти, виконували вручну. Та кожен заклад, або підрозділ, мають різний підхід у реалізації того, чи іншого засобу автоматизації. У нашому випадку, основним і можна сказати, ефективним, шляхом реалізації даного програмного засобу, є створення вебдодатку. Даний додаток має наступне призначення:

- можливість забезпечення реєстрації поточної успішності студентів;
- аналіз отриманих даних;
- організувати доступ цих даних до всіх студентів університету.

Основний процес роботи системи відбуватиметься між викладачем і студентом. Розглянемо більш детально, що саме може зробити викладач, у такій системі:

- зареєструвати та/або змінити бал, отриманий студентом на занятті;
- зареєструвати та/або змінити дані, щодо присутності студента на занятті;
- змінити атрибути заняття, такі як вид і дату заняття;
- візуалізувати дані успішності студентів однієї групи/поток, для подальшого аналізу.

У свою чергу студент, у такій системі, може робити наступне:

1) переглянути результати власного навчання, а саме:

а) дані про відвідування занять;

б) бали отримані на лекційних/практичних/семінарських/лабораторних заняттях;

2) візуалізувати дані власного навчання, для подальшого аналізу.

1.3 Аналіз існуючих програмних засобів автоматизованого обліку

На сьогодні існує достатньо різноманітна добірка засобів автоматизованого обліку, і враховуючи розвиток технологій, способів їх реалізації може бути велика кількість.

Так, одне з приватних підприємств "Політек-СОФТ", є автором досить широкого набору програмного забезпечення, яке призначене для закладів вищої освіти (ЗВО) [1], а саме:

- пакет програм "Деканат" – основною функцією якого, є автоматизація планування й обліку навчального процесу;
- програмний модуль "ПС-Журнал успішності-Web" – є розширенням функціональних можливостей до попереднього пакету програм "Деканат".

На рисунку 1.1 наведено частину закладів освіти у місті Києві, які користуються продуктами від даного підприємства. Загалом налічується понад 250 ЗВО, які користуються послугами "Політек-СОФТ".

| | | | | | | |
|----|------|----------|-------|------|---------------------------|------|
| | | | | | ІАЛЦ.045440.004 ПЗ | Лист |
| Зм | Лист | № докум. | Підп. | Дата | | 7 |

Кількість впроваджень програмного забезпечення ПП "Політек-СОФТ" в Київській області складає: 94.

| Місто | Назва вищого навчального закладу | Акредитація | Продукт |
|-------------|---|-------------|--------------------------|
| Київ | Дипломатична академія України при МЗС України | 3-4 | Деканат-С-Університет |
| Київ | Міжнародний інститут ринкових відносин та підприємництва | 3-4 | Деканат-С-Міні |
| Київ | Національний авіаційний університет | 3-4 | ПС-Абітурієнт |
| Київ | Національний університет харчових технологій | 3-4 | Деканат+-Університет(Гр) |
| Київ | ПВНЗ `Київський університет культури` | 3-4 | Деканат+-Університет |
| Київ | ПВНЗ `Київський університет культури` | 3-4 | ПС-Персонал |
| Київ | ПВНЗ `Київський університет культури` | 3-4 | ПС-Абітурієнт |
| Біла Церква | Білоцерківське міське територіальне представництво Відкритого міжнародного університету розвитку людини "Україна" | 3-4 | Бібліограф |
| Бровари | Економіко-технологічний університет | 3-4 | Деканат+-Університет |
| Київ | Європейський університет | 3-4 | Деканат-Р |
| Київ | Інститут підготовки кадрів державної служби зайнятості України (ІПКДСЗУ) | 3-4 | Бібліограф-Web-2007 |
| Київ | Інститут підготовки кадрів державної служби зайнятості України (ІПКДСЗУ) | 3-4 | Деканат-С-Університет |
| Київ | Інститут підготовки кадрів промисловості України (ІПКПУ) | 3-4 | Деканат+ |
| Київ | Київський Інститут Бізнесу і Технологій (КІБІТ) | 3-4 | Деканат-С |
| Київ | Київський національний економічний університет імені Вадима Гетьмана (КНЕУ) | 3-4 | Деканат+-Університет |
| Київ | Київський національний економічний університет імені Вадима Гетьмана (КНЕУ) | 3-4 | Колоквіум |
| Київ | Київський національний торговельно-економічний університет (КНТЕУ) | 3-4 | Колоквіум |
| Київ | Київський національний торговельно-економічний університет (КНТЕУ) | 3-4 | Деканат-С-Міні |
| Київ | Київський національний університет будівництва і архітектури | 3-4 | Деканат-С-Міні |
| Київ | Київський національний університет культури і мистецтв | 3-4 | ПС-Абітурієнт |
| Київ | Київський національний університет культури і мистецтв | 3-4 | Деканат+-Університет |
| Київ | Київський національний університет культури і мистецтв | 3-4 | ПС-Персонал |
| Київ | Київський Університет імені Бориса Грінченка | 3-4 | Деканат+-Університет(Гр) |
| Київ | Київський університет "Східний світ" | 3-4 | Деканат-Міні-Університет |

Рисунок 1.1 – Заклади освіти, які використовують ПЗ підприємства "Політек-СОФТ" у місті Києві

Ще одним прикладом є програмний комплекс "Автоматизована система управління навчальним закладом" [2]. Створено даний комплекс у місті Харкові, у Національному університеті міського господарства імені О. М. Бекетова, і використовується він не тільки у ЗВО цього міста, а й у закладах в Кременчуцькій, Сумській, Вінницькій, Одеській та Київській областях. Цей комплекс забезпечує керування ЗВО в єдиному інформаційному просторі, та складається з модулів, які працюють у середовищі ОС Windows, та вебпорталі.

На рисунку 1.2 зображено сервіс "Електронний журнал", який призначений для обліку студентів у визначеному році та семестрі.

Електронний журнал

Учебный год Семестр

| Дисциплины | Группы |
|--|---|
| Обществознание (Пз), 10с Обществознание (Пз), 11с Теория государства и права (Лк), 21, 22, 22 | Группа 1 Группа 2 Группа 3 |

| № | Ф.И.О. студента | 11 | 12 | 13 | 14 | 17 | 18 | 19 | 20 | 21 | № | Ф.И.О. студента | 11 | 12 | 13 | 14 | 17 | 18 | 19 | 20 | 21 | Доп. баллы | Итог | П.1/К. | Всего | Пер. 1 | Пер. 2 |
|----|-----------------|----|----|----|----|----|----|----|----|----|----|-------------------|----|----|----|----|----|----|----|----|----|------------|------|--------|-------|--------|--------|
| | | 01 | 01 | 01 | 01 | 01 | 01 | 01 | 01 | 01 | | | 01 | 01 | 01 | 01 | 01 | 01 | 01 | 01 | 01 | | | | | | |
| 1 | Студент 1 | | | | | | | | | | 1 | Виктор С. В. | | | | | | | | | | | | | | | |
| 2 | Студент 2 | | | | | | | | | | 2 | Григор О. С. | | | | | | | | | | | | | | | |
| 3 | Студент 3 | | | | | | | | | | 3 | Долженко М. В. | | | | | | | | | | | | | | | |
| 4 | Студент 4 | | | | | | | | | | 4 | Журавский В. О. | | | | | | | | | | | | | | | |
| 5 | Студент 5 | | | | | | | | | | 5 | Коваленко Д. В. | | | | | | | | | | | | | | | |
| 6 | Студент 6 | | | | | | | | | | 6 | Кузнецов А. В. | | | | | | | | | | | | | | | |
| 7 | Студент 7 | | | | | | | | | | 7 | Луган К. С. | | | | | | | | | | | | | | | |
| 8 | Студент 8 | | | | | | | | | | 8 | Макогон Г. С. | | | | | | | | | | | | | | | |
| 9 | Студент 9 | | | | | | | | | | 9 | Макогон Н. В. | | | | | | | | | | | | | | | |
| 10 | Студент 10 | | | | | | | | | | 10 | Сас Д. Г. | | | | | | | | | | | | | | | |
| 11 | Студент 11 | | | | | | | | | | 11 | Таранов А. С. | | | | | | | | | | | | | | | |
| | | | | | | | | | | | 12 | Удовинченко Т. В. | | | | | | | | | | | | | | | |
| | | | | | | | | | | | 13 | Усанов В. В. | | | | | | | | | | | | | | | |
| | | | | | | | | | | | 14 | Хачатрян А. А. | | | | | | | | | | | | | | | |
| | | | | | | | | | | | 15 | Шабельник В. А. | | | | | | | | | | | | | | | |
| | | | | | | | | | | | 16 | Шалкина А. О. | | | | | | | | | | | | | | | |

Рисунок 1.2 – Сервіс "Електронний журнал"

Наступний програмний засіб який буде розглянуто, є відомим не тільки в Україні, а й за її межами. Мова йде про Moodle (Modular Object-Oriented Dynamic Learning Environment) – модульне об'єктно-орієнтоване динамічне навчальне середовище, яке ще називають:

- системою для керування навчанням/курсами;
- середовищем для віртуального навчання.

Даний засіб використовується не тільки у ЗВО, а й також у загальноосвітніх навчальних закладах, бізнес-навчанні, тощо. На рисунку 1.3 зображено домашню сторінку тестового сервісу Moodle [3].

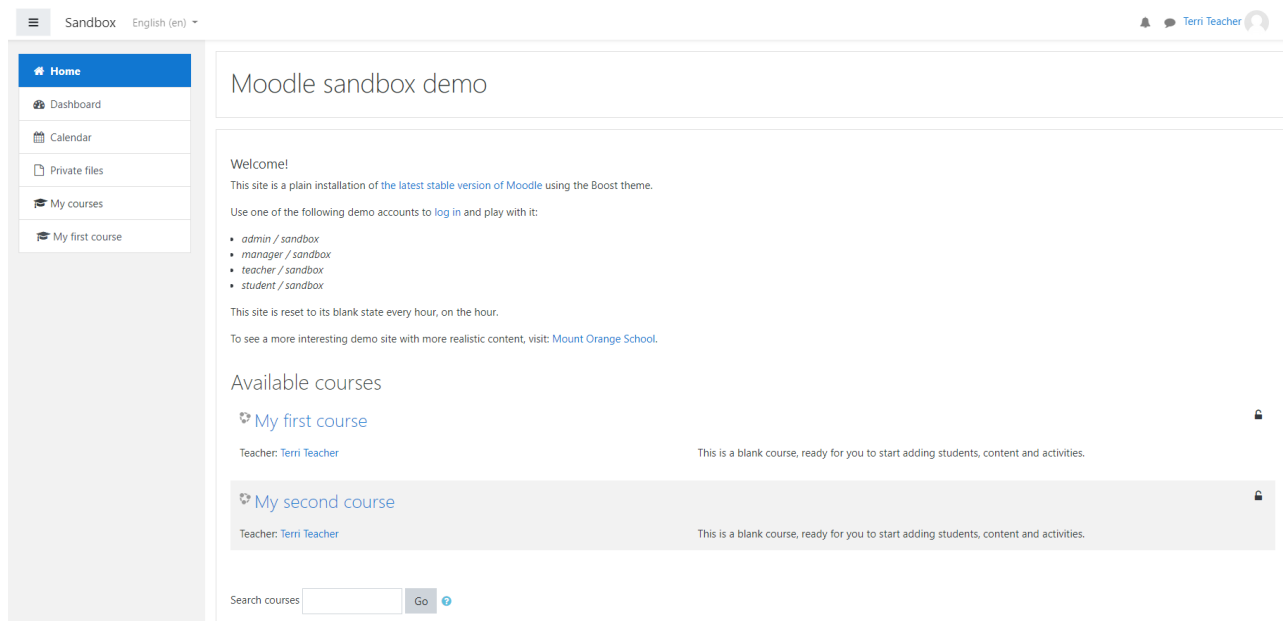


Рисунок 1.3 – Домашня сторінка тестового сервісу Moodle

Також варто зазначити, що дана система є абсолютно безкоштовною і відкритою (Open Source). На даний момент вона налічує 129 мільйонів користувачів у всьому світі й продовжує свій розвиток. І це є однією з переваг даного сервісу, у порівнянні з двома попередніми. Не дивлячись на таку кількість користувачів, з рисунку 1.4 можемо бачити, що документація цього сервісу перекладена лише на 9 мов, і це робить її не зовсім придатною для ЗВО в Україні [4].

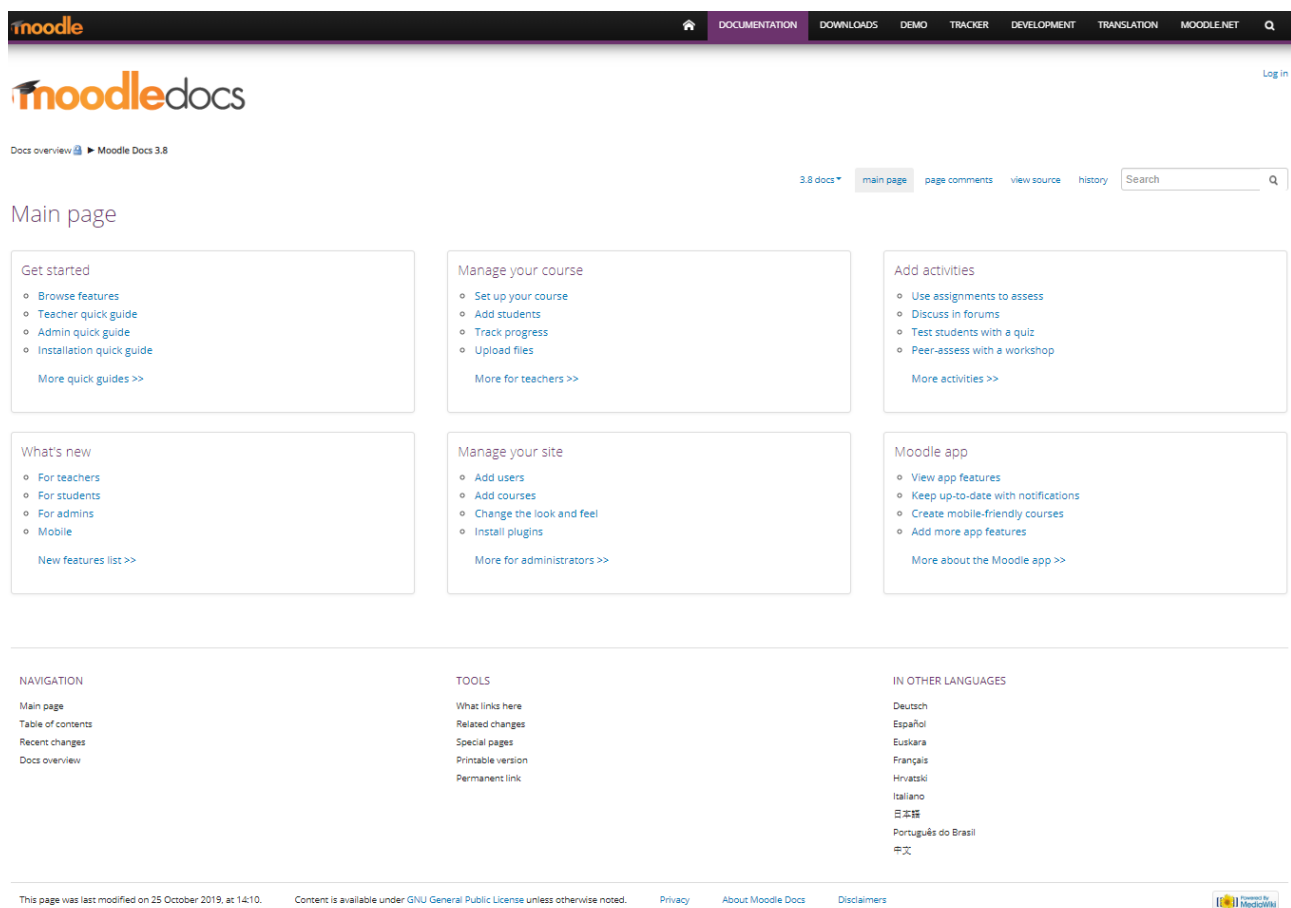


Рисунок 1.4 – Сторінка документації сервісу Moodle

1.4 Обґрунтування теми дипломного проєкту

Всі розглянуті засоби, тим чи іншим чином мають певні недоліки. Основним з них є відсутність якогось одного сервісу, який має повністю український інтерфейс, в тому числі документацію.

Кожен з розглянутих сервісів, дозволяє викладачам та студентам, створювати, додавати, редагувати, видаляти певні дані, в залежності від користувача. І саме тому було вирішено у даному дипломному проєкті, реалізувати аналіз даних, який може допомогти викладачу краще зрозуміти де студенти відстають/переганяють навчальний план, і в свою чергу студенти

| | | | | | | |
|----|------|----------|-------|------|---------------------------|------|
| | | | | | ІАЛЦ.045440.004 ПЗ | Лист |
| Зм | Лист | № докум. | Підп. | Дата | | 11 |

можуть зрозуміти, яка дисципліна, яка контрольна/лабораторна робота далися студенту важко. Аналіз даних є необхідною частиною, адже людський мозок, краще розуміє інформацію яку візуалізовано, ніж ту, яка представлена у вигляді таблиці. Також, варто зазначити, що процес візуалізації є досить гнучким, і користувач може обирати як саме відобразити, ту, чи іншу інформацію.

Висновок до розділу

У цьому розділі було розглянуто причини необхідності автоматизованого обліку, та основні функції, які необхідні для роботи з такою системою. Також було розглянуто та проаналізовано існуючі системи автоматизованого обліку, та обґрунтовано тему дипломного проєкту. У наступному розділі будуть розглянуті та проаналізовані обрані засоби для безпосереднього створення такої системи, та сама архітектура такої системи.

| | | | | | | |
|----|------|----------|-------|------|---------------------------|------|
| | | | | | ІАЛЦ.045440.004 ПЗ | Лист |
| Зм | Лист | № докум. | Підп. | Дата | | 12 |

2 АНАЛІЗ ТЕХНОЛОГІЙ ДЛЯ РОЗРОБКИ ПРОГРАМНИХ ЗАСОБІВ АВТОМАТИЗОВАНОГО ОБЛІКУ ТА АНАЛІЗУ ДАНИХ

2.1 Обґрунтування обраного типу прикладного програмного забезпечення для розробки програмних засобів автоматизованого обліку

У попередньому розділі нами було розглянуто приклади існуючих систем автоматизованого обліку. Кожна з них є вебдодатком, і розроблений у даному дипломному проєкті програмний засіб, також є ним.

Вебдодаток – це прикладне програмне забезпечення, яке переважно створено на основі клієнт-серверної моделі. Відповідно клієнтом виступає браузер, а сервером – вебсервер (рисунок 2.1) [5].

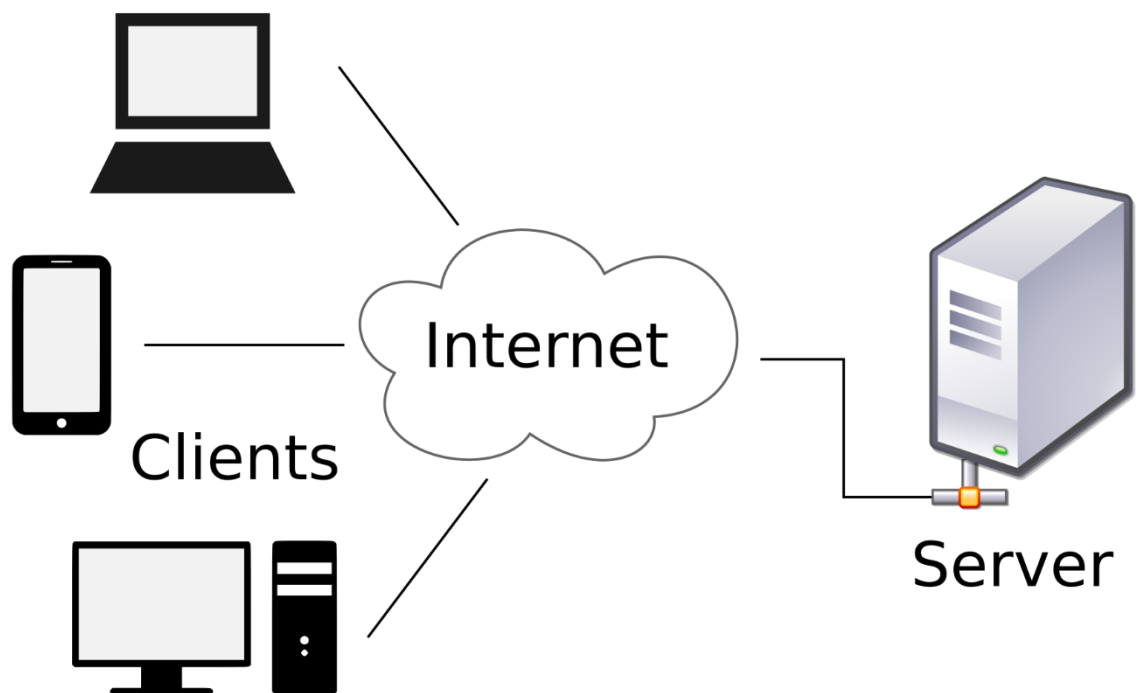


Рисунок 2.1 – Модель клієнт-сервер

Даній моделі надають перевагу при створенні розподілених мережних додатків, та складається вона з наступних компонентів:

- набору серверів, які передбачають надання інформації програмам, котрі до них звертаються;
- набору клієнтів, які передбачають використання сервісів, котрі надаються серверами;
- мережа, яка забезпечує роботу між клієнтами та серверами.

Також варто зазначити, що як сервери, так і клієнти, є незалежними один від одного. Прикладами комп'ютерних додатків, які використовують клієнт-серверну модель, є електронна пошта та звісно всесвітня павутина (World Wide Web - WWW). Це зумовлено наступним рядом причин:

- відсутність потреби для завантаження спеціальних програм, адже всі комп'ютери мають браузер, який виступає клієнтом;
- гнучкий доступ, адже є можливість користуватися ним у будь-якій точці на Землі, де є Інтернет з'єднання;
- відсутність залежності від технічних характеристик певного пристрою;
- відносно просте налаштування;
- збільшення розміру сховища даних, адже з розвитком хмарних технологій, можна сказати, маємо нескінченний віртуальний простір для збереження даних.

В свою чергу, вебдодатки, діляться на рівні, де кожен рівень має свою роль. І найбільш уживаною є трирівнева архітектура (рисунок 2.2).

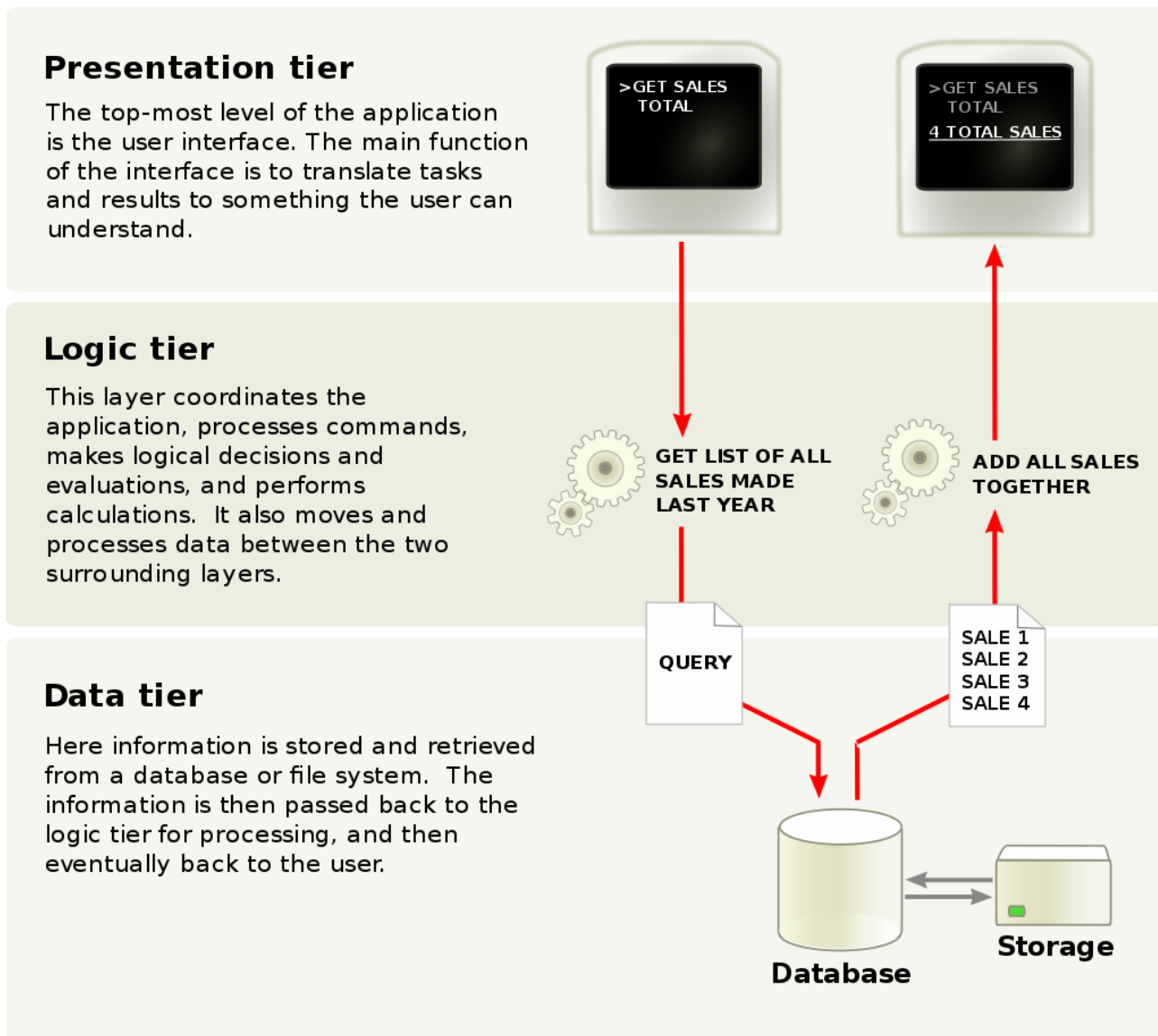


Рисунок 2.2 – Трирівнева клієнт-серверна архітектура

На рівні представлення реалізовано графічний інтерфейс, за допомогою якого, користувач взаємодіє з клієнтом. Логічний, або як його ще називають, середній рівень, контролює функціональність програми, і здійснює обробку даних, отриманих з рівня представлення. Наступний рівень, рівень даних, включає механізм збереження даних, та рівень доступу до даних.

Та існує інший архітектурний стиль Peer-to-peer (P2P), що перекладається як рівний до рівного. І мережа рівноправних вузлів є основою даного стилю [6], що зображено на рисунку 2.3.

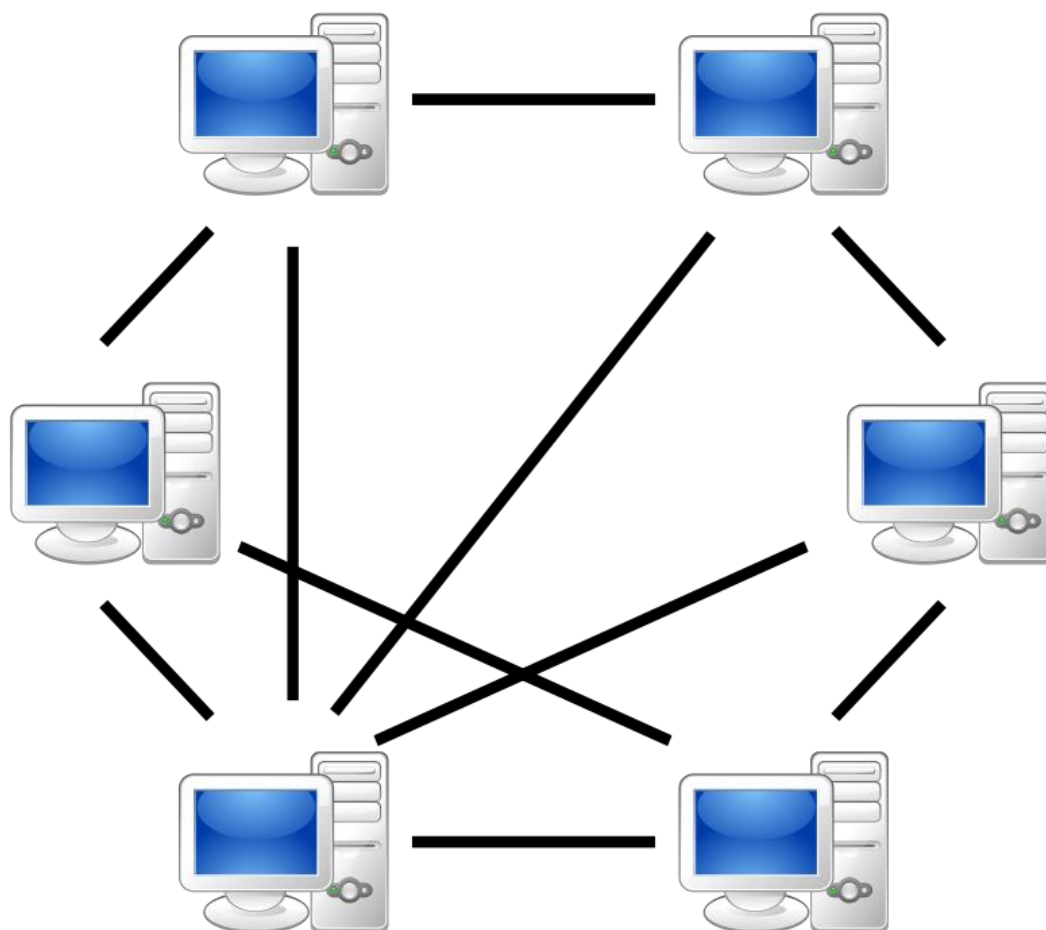


Рисунок 2.3 - Приклад P2P мережі

Отже, на відміну від попередньо розглянутої клієнт-серверної моделі, даний архітектурний стиль не має поняття клієнта та сервера. Як можна дізнатися з назви, поняття, яке є основним для цієї моделі, є рівні вузли, які функціонують як клієнт, так і сервер одночасно. І тут є одна значуща перевага цієї організації - вона дає можливість збереження працездатності мережі, при будь-якій конфігурації для доступних учасників. Окрім цього P2P мережа має наступні переваги:

- зменшення або розподіл вартості;
- удосконалена надійність;
- можливість роботи більш автономно;

- конфіденційність;
- динамічність;
- об'єднання ресурсів.

Та не дивлячись на цей ряд переваг, клієнт-серверна модель використовується набагато більше ніж P2P. Клієнт-серверна модель є правильним варіантом для реалізації поставленої задачі, адже вона:

- більш стабільна та безпечна, ніж P2P;
- є централізованою мережею, у порівнянні з P2P;
- може використовуватися як для мережі малого та великого розмірів, тоді як, мережа P2P, здебільшого використовується для мереж малого розміру;
- є простою в управлінні.

На рисунку 2.4 можемо наглядно побачити різницю у роботі двох архітектурних стилів:

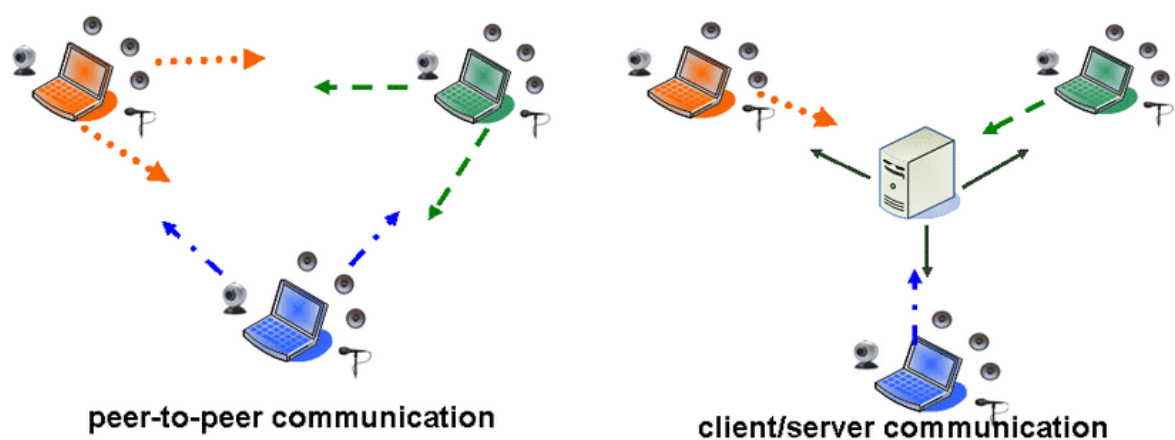


Рисунок 2.4 – Порівняння клієнт-серверної та P2P мереж

2.2 Порівняльний аналіз програмних засобів для розробки серверних додатків

Розглянемо засоби, які були використані для розробки серверних додатків, для систем які було наведено у попередньому розділі. Всі з них використовували мову серверних сценарії Hypertext Preprocessor (PHP) [7], яка наразі є найпопулярнішою (рисунок 2.5).

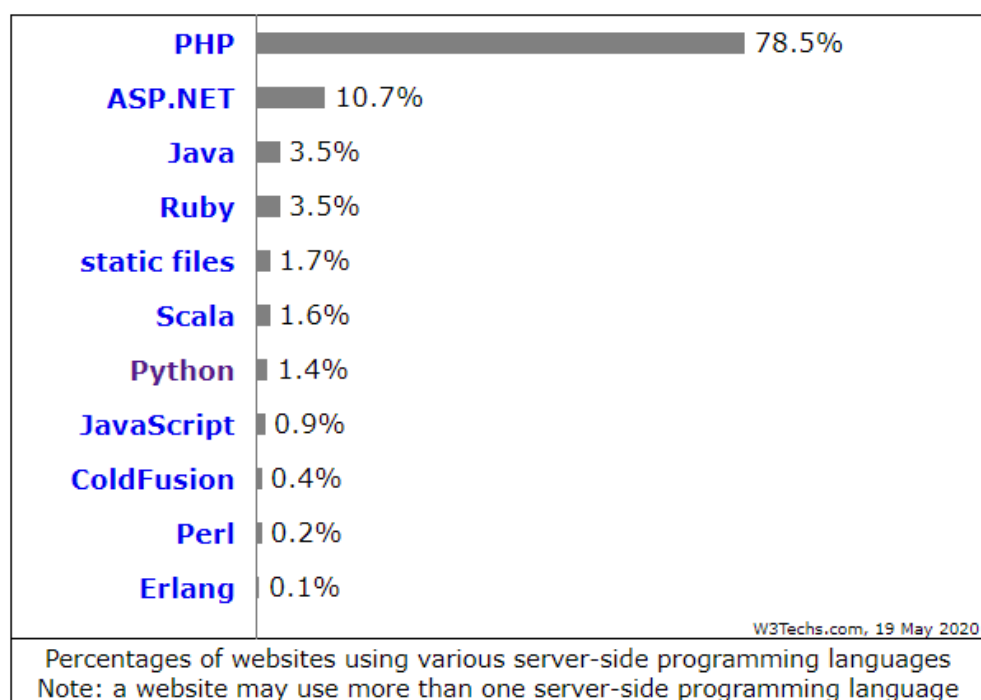


Рисунок 2.5 – Статистика технологій для розробки серверних додатків

Це зумовлено тим, що PHP, є відносно простою мовою в плані синтаксису. Вона є простою для завантаження, і не сильно залежить від ОС. Багато вебдодатків створювалися за допомогою PHP, і з часом тільки удосконалювалися, що стало причиною створення великої спільноти, яка підтримує і розвиває мову, адже вона є відкритою, що дає розробникам можливість зробити певний внесок у її розвиток.

Та для даного дипломного проєкту було вирішено використовувати іншу мову програмування - Python. Згідно досить відомому індексу популярності мови програмування (Popularity of Programming Language - PYPL), мова Python залишається безсумнівним лідером [8], і має найбільший приріст популярності, серед п'ятірки лідерів (рисунок 2.6).

| Worldwide, May 2020 compared to a year ago: | | | | |
|---|--------|------------|---------|--------|
| Rank | Change | Language | Share | Trend |
| 1 | | Python | 31.17 % | +4.3 % |
| 2 | | Java | 17.75 % | -2.4 % |
| 3 | | Javascript | 7.99 % | -0.3 % |
| 4 | | C# | 7.05 % | -0.2 % |
| 5 | | PHP | 6.09 % | -1.0 % |

Рисунок 2.6 – Рейтинг мов програмування станом на травень 2020 року

Мова Python стала популярною відносно недавно, і за цей час встигла показати себе майже у кожній сфері створення програмного забезпечення. Вона є простою для вивчення, що стало однією з причин цього розвитку. Та зараз Python є провідним інструментом у роботі з даними [9], та веброзробці, згідно опитувань минулого року від компанії JetBrains (рисунок 2.7).

What do you use Python for?

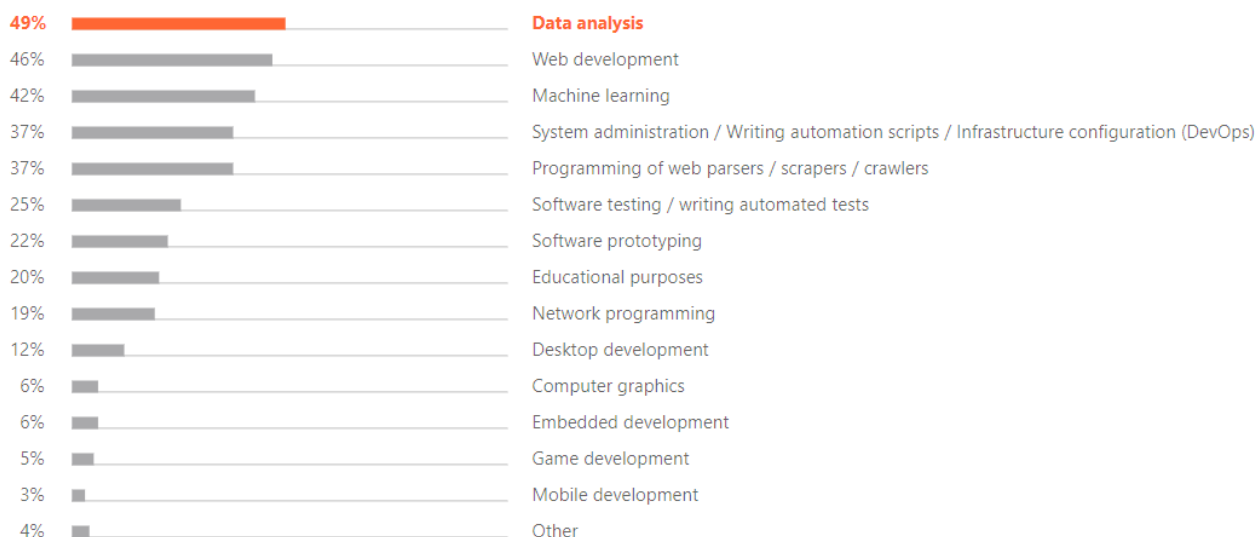


Рисунок 2.7 – Рейтинг напрямків, у яких найбільше застосовується Python

Даний рейтинг ще раз підтверджує, актуальність та вірність обраних засобів мови Python для виконання даного дипломного проєкту – аналіз даних та веброзробка. Для серверної частини буде використаний потужний вебфреймворк Django. Знову ж таки згідно опитуванню JetBrains, провідної компанії з розробки програмного забезпечення, можна пересвідчитись у актуальності обраних засобів. На наступному рисунку можна побачити рейтинг засобів для веброзробки (рисунок 2.8).

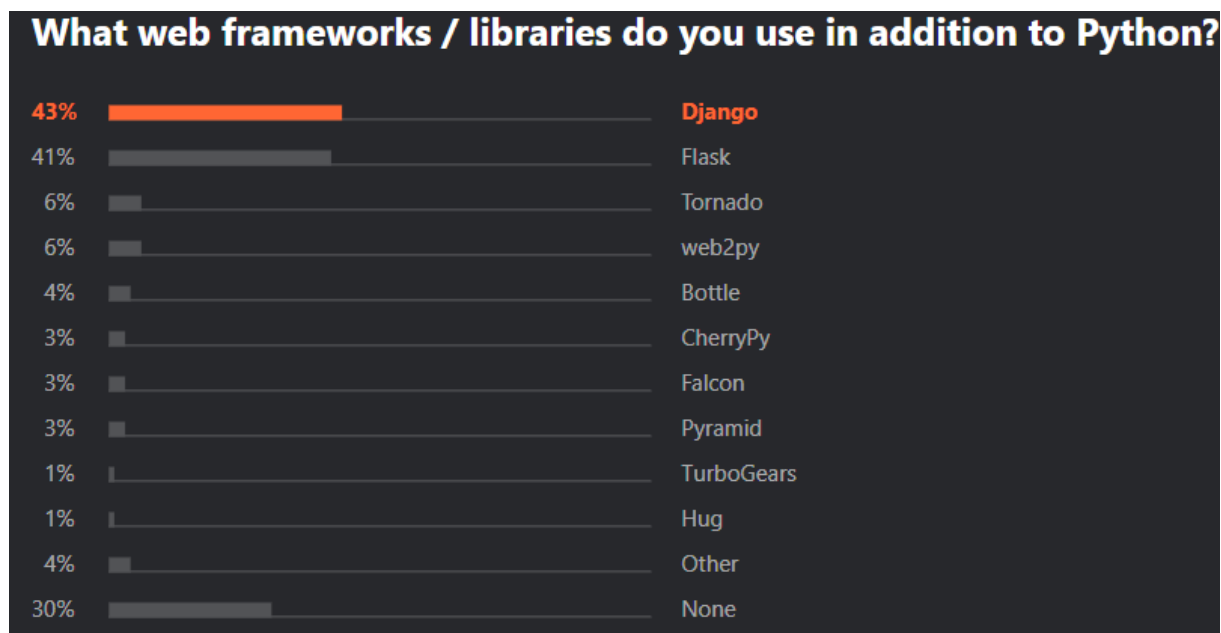


Рисунок 2.8 – Рейтинг вебфреймворків Python

Фреймворк Django буде виконувати роботу на логічному рівні нашої трирівневої архітектури. Не дивлячись на кількість існуючих вебдодатків, які використовують мову PHP, зараз вона втрачає свою популярність, і витісняється іншими технологіями зі світового ринку, такими як Django, які з кожним роком, стають більш уживаними. На рисунку 2.9 показано різницю між фреймворком Django, та мовою PHP [10].

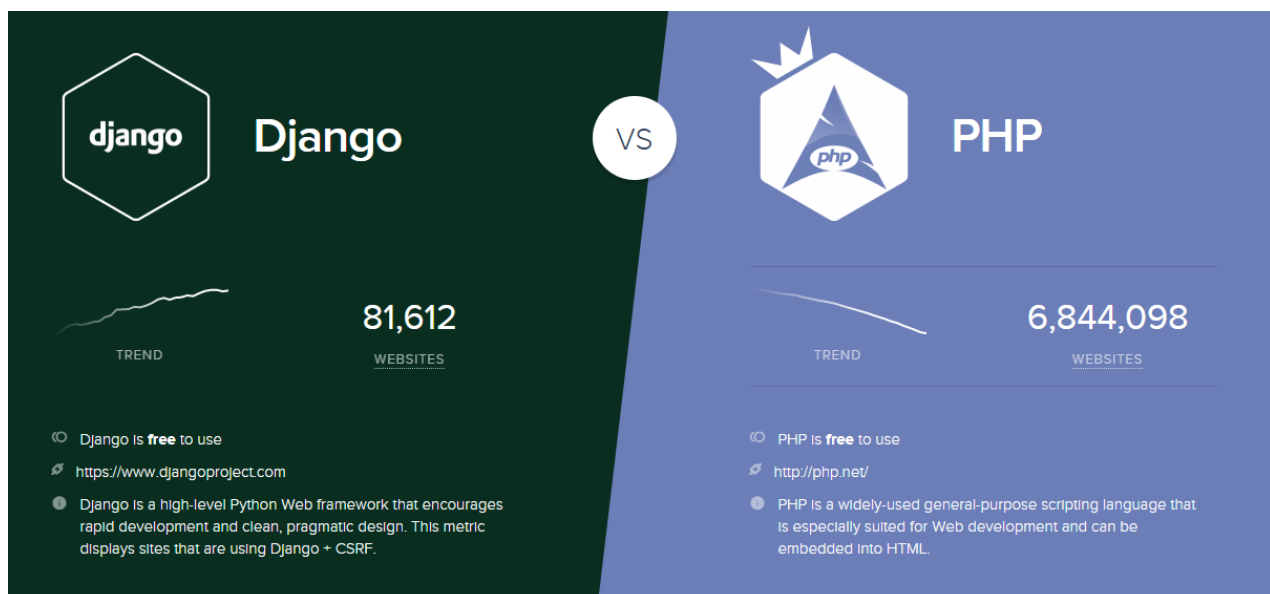


Рисунок 2.9 – Порівняння Django та PHP

Написаний на чистому Python, Django має чисту "пітонічну" структуру. Він почав як Модель-Вид-Контролер (Model-View-Controller - MVC) фреймворк, і зараз ця концепція існує досі у поточній версії [11]. Ця архітектура дозволяє розробникам змінювати візуальну частину програми та частину ділової логіки окремо, не впливаючи один на одного. Але насправді розробники зазвичай називають архітектуру Django як Модель-Вид-Шаблон (Model-View-Template - MVT). На рисунку 2.10 зображено три рівні, які в свою чергу відповідають за різні речі, та можуть використовуватися незалежно.

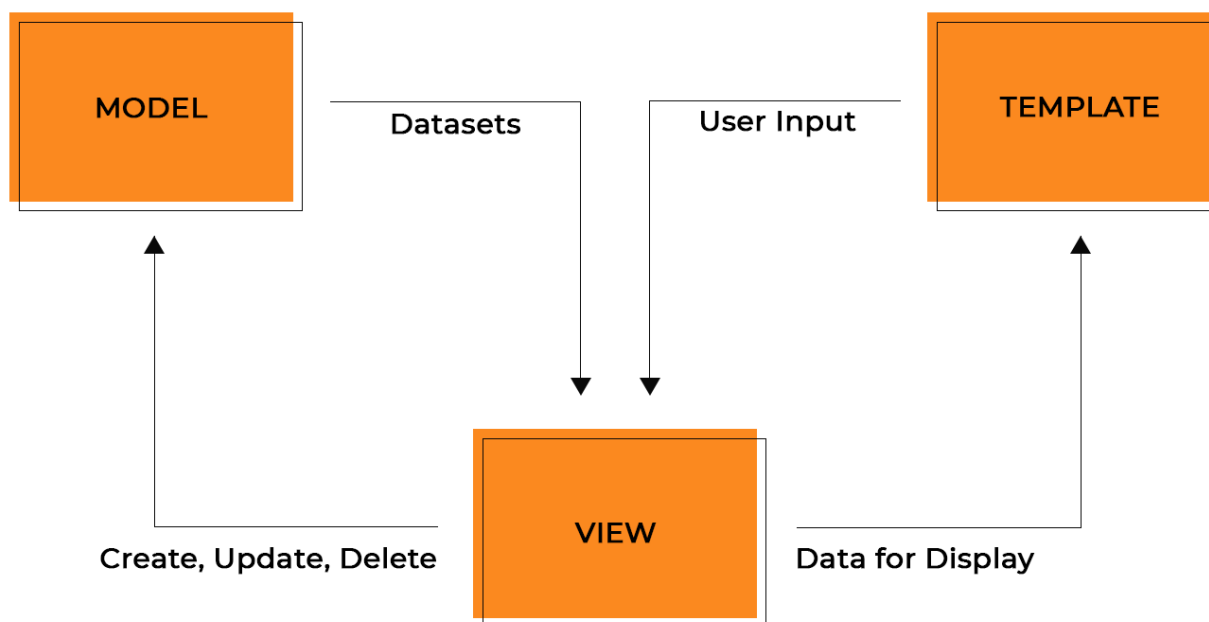


Рисунок 2.10 – Архітектура Django

Як зазначено у документації Django, модель це "єдине, точне джерело інформації про ваші дані. Воно містить найважливіші поля та поведінку даних, які ви зберігаєте". Як правило, кожна модель відображається в одній таблиці баз даних. Навряд чи є додаток без бази даних, і Django офіційно підтримує чотири:

- PostgreSQL;
- MySQL;
- SQLite;
- Oracle.

Моделі містять дані та представлені атрибутами (полями). Оскільки модель є простим класом Python, вона не знає нічого про інші рівні Django. Зв'язок між цими рівнями можливий лише через інтерфейс програмування додатків (Application Programming Interface - API). Моделі містять ділову логіку, спеціальні методи, властивості та інші речі, пов'язані з роботою з даними. Також моделі дають розробникам можливість створювати, читати, оновлювати та видаляти об'єкти (набори даних) у вихідній базі даних.

Модуль перегляду "view" виконує завдання дерева: він приймає HTTP-запити (Hyper Text Transfer Protocol (HTTP) – протокол передачі гіпертекстових документів), застосовує ділову логіку, надану класами та методами Python, та надає відповіді HTTP на запити клієнтів. Іншими словами, цей модуль отримує дані з моделі та надає кожному шаблону доступ до конкретних даних, що відображаються, або попередньо обробляє дані.

Django має потужну систему шаблонів і власну мову розмітки з багатьма інструментами. В свою чергу шаблони – це файли з HTML-кодом (HyperText Markup Language (HTML) – мова розмітки гіпертексту), які використовуються для візуалізації даних. Вміст цих файлів може бути статичним або динамічним. Оскільки шаблон немає ділової логіки, він створений лише для представлення даних. На рисунку 2.11 зображено приклад такого шаблону.

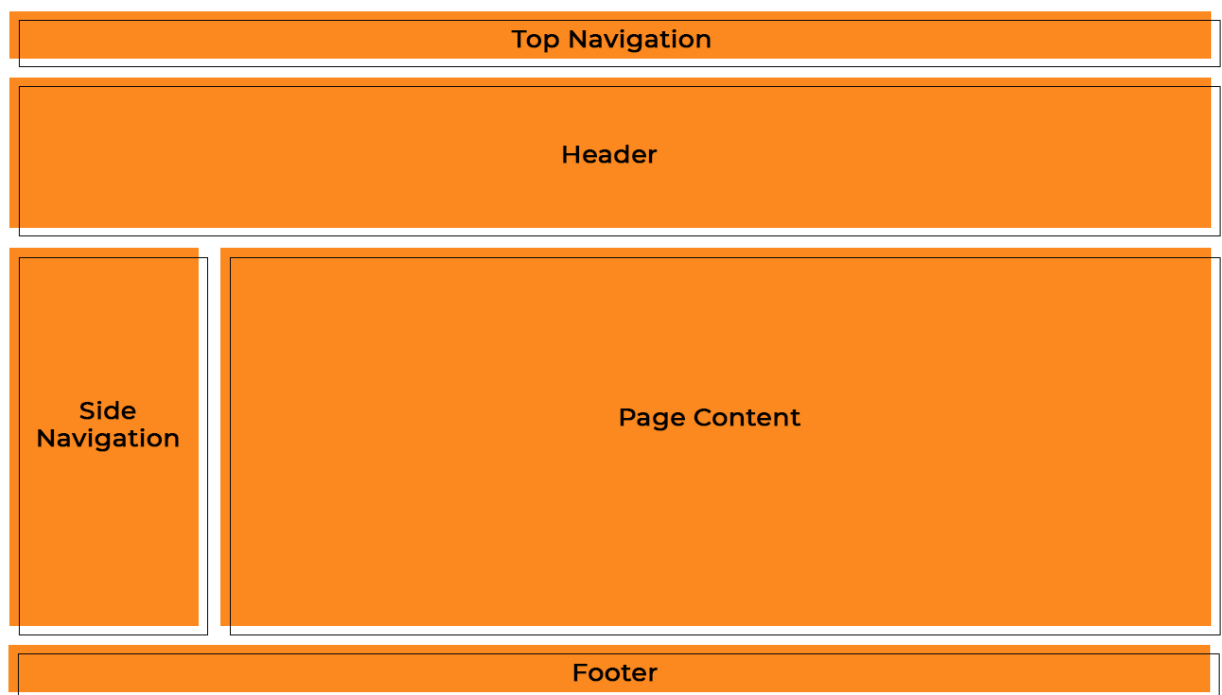


Рисунок 2.11 – Шаблон Django

Походження назви фреймворку є ключовим для розуміння його багатогранності. Свою назву фреймворк Django зобов'язаний джазовому

гітаристові Джанго Рейнхардту, який зміг грати на гітарі навіть коли два пальці були паралізовані після аварії. Таким же чином, фреймворк Django може взяти на себе численні завдання. Його можна використовувати для створення:

- системи керування зв'язками з клієнтами (client relationship management - CRM systems);
- системи керування вмістом для внутрішнього та комерційного використання (content management systems - CMS);
- комунікаційні платформи;
- платформи для бронювання;
- платформи управління документами;

Крім усього іншого, фреймворк Django, чудово підходить для:

- генераторів на основі алгоритму;
- системи верифікації;
- системи фільтрації з правилами, які динамічно змінюються, та розширеними параметрами;
- рішення задач із аналізу даних та складних обчислень;
- машинного навчання.

На рисунку 2.12 можна побачити використання фреймворку Django серед країн усього світу [12]. Також варто зазначити наявність України у списку країн-лідерів, що свідчить про потребу використання сучасних технологій на території нашої країни, що є важливим аспектом для розвитку.



Рисунок 2.12 – Використання фреймворку Django серед країн світу

Розглянемо переваги даного фреймворку, серед яких:

- багата екосистема;

Розробники кажуть: "Читайте Django як систему". Під цим вони мають на увазі, що існує багато сторонніх додатків, які постачаються з Django. Ці програми можуть бути інтегровані залежно від вимог проєкту.

- зрілість;

Django існує вже 11 років, і за цей час пройшов етапи значного вдосконалення. Багато речей було доведено до досконалості, та відповідно додано багато нового. Головним є те, що, будь-які питання по роботі з Django, вже мають відповідь, адже пристрасна спільнота з тисячі людей, вже вирішили ці питання.

- панель адміністратора за замовчуванням;

Панелі адміністраторів розроблені, щоб керувати нашою програмою. Ця панель формується автоматично з коду Python, і ми не витрачаємо жодного часу на її створення, тоді як створення такої панелі вручну, зайняло б багато часу, і було б абсолютно недоречним. Завдяки стороннім програмам, є можливість для максимальної кастомізації панелі адміністрування. Крім того, Django дозволяє змінювати інтерфейс за допомогою сторонніх обгортки, і додавати панелі приладів, унікальні для потреб розробника.

- підходить для пошукової оптимізації сайту (search engine optimization - SEO);

Python відомий через те що має читабельний код, і це перевага, якщо розробник хоче, щоб його сайт займав високі позиції в результатах пошуку. За допомогою Django можна генерувати URL-адреси (Uniform Resource Locator (URL) - єдиний вказівник на ресурс) та посилання вебсайтів, які можуть бути зчитані, використовуючи найрелевантніші ключові слова та найкращі практики оптимізації пошукових систем.

- має можливість підключення додаткових плагінів;

Django є підключеним сам за своєю природою, і може бути розширений плагінами. Плагіни, в свою чергу, це програмні компоненти, які дозволяють розробникам додавати певну функцію до програми, залишаючи багато можливостей для налаштування. Є сотні плагінів, які можуть дозволити розробнику додати Google Карти, створювати права доступу різної складності, або підключитися до Stripe для обробки платежів. Якщо є необхідність у масштабуванні проєкту у майбутньому, можна відключати деякі компоненти, додавати нові, які будуть відповідати поточним потребам.

- бібліотеки;

Кожна мова програмування має власний набір бібліотек для вирішення загальних задач. Бібліотека програмного забезпечення включає в себе, попередньо записаний код, класи, процедури, сценарії, дані конфігурацій, тощо. Як правило, до програм додають бібліотеки, для надання більшої функціональності або для автоматизації процесу без написання нового коду вручну. Django дозволяє розробникам використовувати бібліотеки під час створення будь-якого проєкту. Деякі популярні бібліотеки включають структуру Django REST, яка відповідає за створення API; Django CMS, який призначений для керування вмістом вебсайту; Django-allauth, що представляє собою інтегрований набір програм Django для аутентифікації, реєстрації, керування обліковим записом та аутентифікації сторонніх (соціальних) облікових записів.

- Object-relational mapping (ORM) - об'єктно-реляційне відображення.

Django цінується за підтримку ORM, що допомагає розробникам взаємодіяти з базами даних. У свою чергу ORM – це бібліотека, яка автоматично передає дані, які зберігаються в базах даних, таких як PostgreSQL і MySQL, в об'єкти, які зазвичай використовуються в коді програми.

Здатність ORM Django збирати інформацію прискорює розробку вебдодатків та допомагає розробникам за короткий час створити працюючі прототипи. Розробникам не обов'язково знати мову, яка використовується для передачі даних для обробки даних.

Розглядаючи клієнт-серверну модель, було розглянуто що, в ролі сервера виступає вебсервер. Та у нашому випадку буде використовуватися вбудований сервер для розробки від самого фреймворку Django, приклад якого, зображено на рисунку 2.13.

| | | | | | | |
|----|------|----------|-------|------|---------------------------|------|
| | | | | | ІАЛЦ.045440.004 ПЗ | Лист |
| Зм | Лист | № докум. | Підп. | Дата | | 28 |

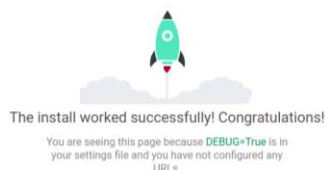


Рисунок 2.13 – Сторінка серверу для розробки Django по замовчуванню

Цей сервер дозволяє запускати код швидше, без додаткових витрат часу на налаштування виробничого серверу. Коли він запущений, він автоматично перевіряє зміни у коді, і перезапускається самостійно, звільняючи розробника від потреби ручного перезапуску серверу. Та певні зміни, такі як додавання нових файлів у проєкт, вимагають ручного перезапуску сервера.

2.3 Порівняльний аналіз систем керування базами даних (СКБД)

Розглянемо системи керування базами даних (СКБД), які використовувалися для попередньо реалізованих систем автоматизованого обліку. Насамперед, варто зазначити, що кожна з далі розглянутих СКБД є реляційною. Відповідно, існують нереляційні СКБД, які не є пріоритетними для роботи з такими системами. У нереляційних СКБД, або NoSQL базах даних, дані зберігаються у вигляді документів, пари "ключ-значення" та графів. І при цьому мають можливість горизонтального масштабування, що є безсумнівною

| | | | | | | |
|----|------|----------|-------|------|---------------------------|------|
| | | | | | ІАЛЦ.045440.004 ПЗ | Лист |
| Зм | Лист | № докум. | Підп. | Дата | | 29 |

перевагою у порівнянні з реляційними СКБД. Та для розробки системи яка створювалася в цьому дипломному проєкті, та інших систем з обліковими записами, правильним вибором буде використання саме реляційних СКБД, адже вони створені саме для вирішення такого роду задач.

Більшість з попередньо розглянутих систем використовували СКБД MySQL, яка теж є дуже вживаною в наш час, і разом з мовою PHP, є зв'язкою, яка зараз підтримує близько 90% існуючих вебдодатків. І як можемо бачити з рисунку 2.14, MySQL є пріоритетною базою для 60% опитаних [13].

Which databases have you used in the last 12 months?

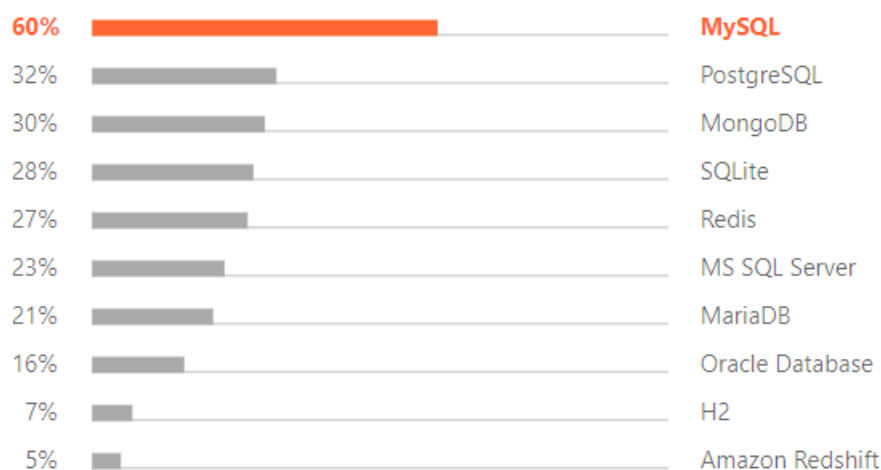


Рисунок 2.14 – Рейтинг найбільш уживаних СКБД

Також можемо бачити, що другу позицію займає PostgreSQL – СКБД, яка була використана у рамках даного дипломного проєкту. Перш за все це зумовлено тим, що вона є пріоритетною при роботі з попередньо розглянутим вебфреймворком Django, та є єдиною СКБД, яка повністю підтримується цим фреймворком. Налаштування Django для роботи з базою даних PostgreSQL є досить простим, і для цього необхідна бібліотека Python - psycopg2. Розглянемо основні відмінності між PostgreSQL та MySQL [14]:

- керування;

Модель керування технологією має унікальні переваги й у випадку MySQL та PostgreSQL демонструє значну різницю. Проєкт MySQL має вихідний код, який є відкритий для широкого загалу за ліцензією GNU, а також декілька фірмових угод. Зараз він належить Oracle Corp. І повинен пропонувати різні платні версії для комерційного використання. В свою чергу PostgreSQL розроблений групою PostgreSQL Global Development і доступний під ліцензією з відкритим кодом, аналогічно ліцензії MIT.

- відповідність мові структурованих запитів (Structured Query Language - SQL);

Стандарт відповідності SQL - це сукупність правил, яких повинна дотримуватися база даних під час реалізації всіх настанов і стандартів SQL. Відповідність SQL є суттєвим обмеженням для компаній, які бажають працювати з неоднорідними базами даних для свого проєкту (програми). PostgreSQL є найбільш сумісною з SQL, оскільки вона відповідає 160 з 179 основних функцій стандарту SQL, а також ряду додаткових функцій. З іншого боку, MySQL частково сумісна з SQL, оскільки не реалізує повний стандарт SQL. Однак вона надає багато корисних функцій, що не належать до SQL, таких як "відсутність обмежень CHECK".

- підтримувані платформи;

Обидві системи PostgreSQL і MySQL можуть працювати на Solaris, Windows Operating System, Linux і ОС X. ОС X розроблена компанією Apple Inc., Solaris продукт компанії Oracle, Windows ОС - продуктом Microsoft, а Linux - операційною системою з відкритим кодом. PostgreSQL також підтримує ОС HP-UX, розроблену технічним гігантом Hewlett-Packard, а також Unix ОС з відкритим кодом. На

відміну від цього, MySQL надає підтримку для відкритої програми FreeBSD.

- підтримка мов програмування;

Якщо система підтримує різні мови програмування, вона допомагає розробникам з різним попереднім досвідом виконувати завдання мовою, яка для них є знайомою, тим самим підвищуючи ефективність. Оскільки сервер підтримує багато мов програмування для функцій бази даних, розробники можуть легко вирішити, виконувати завдання в клієнті чи сервері. Підтримка додаткових мов програмування дає розробникам більше можливостей. PostgreSQL написаний на мові C та підтримує ряд мов програмування, серед яких найбільш помітні C/C++, Delphi, JavaScript, Java, Python, R, Tcl, Go, Lisp, Erlang та .Net. MySQL написаний на C і C++ і розширює підтримку на C/C++, Erlang, PHP, Lisp, Go, Perl, Java, Delphi, R і Node.js.

- безпека;

Захищеність СКБД глибоко впливає на її надійність і стає однією з головних характеристик, яку потрібно шукати в системі. Безпека бази даних вказує на колективне використання процесів та інструментів для захисту СКБД або бази даних від загроз та атак з нелегітимних джерел. MySQL дуже безпечна і включає в себе безліч функцій безпеки. Вона виконує протоколи безпеки на основі списків контролю доступу (Access Control Lists - ACL) для операцій користувача, таких як з'єднання та запити. PostgreSQL пропонує вбудовану підтримку SSL для з'єднань для шифрування комунікацій клієнт/сервер. Більше того, вбудоване вдосконалення під назвою SE-PostgreSQL, яке забезпечує додаткові можливості контролю доступу на основі політики SELinux.

- методи доступу;

І MySQL, і PostgreSQL підтримують усі стандарти та мають кілька загальних методів доступу, до яких належать JDBC, ODBC та

ADO.NET. JDBC - це API для доступу до бази даних для програмування мовою Java, тоді як ODBC - це стандартний API, що використовується для доступу до баз даних. ADO.NET - це набір API, який розробники використовували для отримання доступу до даних на основі XML. Рідну бібліотеку C та потокові API також можуть використовуватися для доступу до PostgreSQL.

– реплікація;

Реплікація бази даних відноситься до процесу електронного копіювання даних з одного комп'ютера або сервера на інший, що дозволяє всім користувачам мати доступ до тієї ж інформації. База даних може використовувати численні методи для зберігання надлишків даних у кількох вузлах. MySQL використовує реплікацію master-master, в якій кожен вузол є головним і має доступ для оновлення даних. Однак і PostgreSQL, і MySQL можуть виконувати реплікацію ведучого-підлеглого, в якій один вузол є головним і керує зберіганням даних іншими вузлами. Інші типи реалізації можуть бути застосовані на практиці в PostgreSQL за допомогою сторонніх розширень.

– продуктивність;

PostgreSQL підходить для використання у великих системах, де дані повинні бути аутентифіковані та швидкості читання/запису є критичними для успіху. Крім того, вона також підтримує ряд покращення продуктивності, які доступні лише у фірмових рішеннях, включаючи одночасність без блоків читання, SQL-сервер та підтримку даних Geospatial серед багатьох інших. Взагалі, PostgreSQL найкраще підходить для систем, які потребують виконання складних запитів, або зберігання даних та аналізу даних.

MySQL - це перший вибір для тих вебпроектів, яким потрібна база даних лише для транзакцій з даними. Вона надзвичайно добре працює

в системах онлайн аналітичної обробки (OnLine Analytical Processing - OLAP) та онлайн обробки транзакцій (OnLine Transaction Processing - OLTP), де необхідні лише високі швидкості читання. Однак MySQL почне слабшати після того, як буде підкреслено великим навантаженням або складні запити. Хоча ми обговорюємо, де ці системи можуть бути використані, фактичну ефективність СКБД можна виміряти лише шляхом оцінки показників можливих сценаріїв, оскільки це значною мірою залежить від вимог та характеру проекту, для якого він використовується.

- підтримка спільноти.

Вона важлива для вдосконалення будь-якої системи баз даних. PostgreSQL має дуже активну спільноту, яка постійно допомагає вдосконалювати існуючі функції, а її винахідливі учасники, які роблять вагомий внесок у розвиток, докладають усіх зусиль для того, щоб база даних залишалася найдосконалішою базою даних з найновішими функціями та максимальною безпекою. MySQL також має велику спільноту та учасників, які зосереджуються на підтримці існуючих функцій, тоді як деякі нові функції з'являються періодично.

Безпосередньо для роботи з СКБД PostgreSQL, було використано такий програмний засіб, як pgAdmin. Завдяки графічному інтерфейсу користувача, він дає змогу більш ефективно працювати з СКБД PostgreSQL. Засіб pgAdmin був створений спеціально для роботи з PostgreSQL, і має наступний ряд переваг:

- підтримує відносно широкий спектр версій PostgreSQL;
- є кросплатформним;
- "вміє" працювати з об'єктами та кодуваннями, які присутні у PostgreSQL.

2.4 Обґрунтування обраних засобів для аналізу даних

Одним із завдань даного дипломного проєкту, є реалізація можливості візуалізації даних, і у нашому випадку, це візуалізація даних про успішність студентів. На наступному рисунку (рисунок 2.15) показано рейтинг бібліотек для роботи з даними.

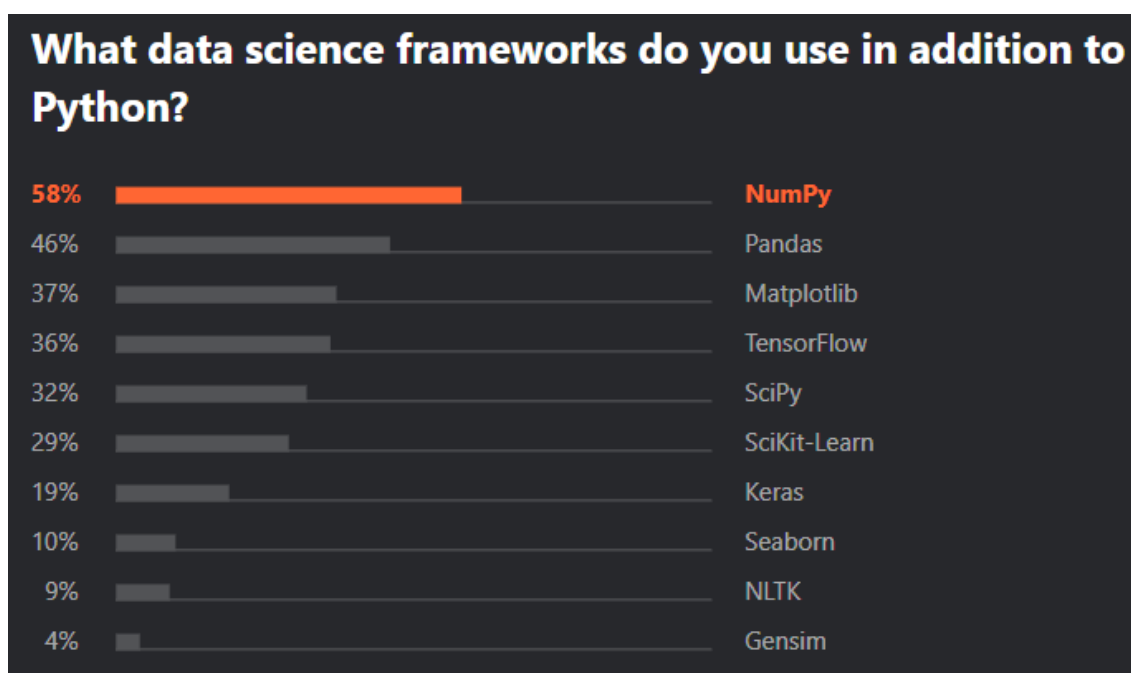


Рисунок 2.15 – Рейтинг бібліотек для роботи з даними

Можемо бачити, що у трійку найбільш уживаних входять NumPy, Pandas та Matplotlib. Та для виконання поставленого завдання було обрано 2 з них, це Pandas та Matplotlib.

Pandas було обрано безпосередньо для роботи з даними, а саме завантаження файлів у .csv форматі для подальшої обробки. Також він має наступні переваги [15]:

- створено безпосередньо для мови Python;

- потребує менше програмного коду для роботи з даними, ніж Python, без додаткових бібліотек;
- досить широкий набір можливостей для роботи з даними;
- орієнтований для роботи з великим об'ємом даних;
- робить роботу з даними більш гнучкою та кастомізованою;
- та звісно має відкритий код, що дає можливість іншим розробникам робити свій вклад у розвиток даної бібліотеки.

В свою чергу бібліотека Matplotlib, була обрана саме для візуалізації даних, які зберігаються у базі даних, які в свою чергу, були попередньо завантажені з файлів у .csv форматі [16]. Тож бібліотека Matplotlib має наступні переваги:

- відносно проста у використанні, адже без проблем може інтегруватися з вище згаданою бібліотекою Pandas;
- дозволяє користувачу майже максимально кастомізувати відображення графіків;
- також має відкритий код.

Отже, можемо зробити висновок, що всі вище описані технології є правильним вибором для реалізації поставленого завдання.

Висновок до розділу

Таким чином, у даному розділі було проведено детальний аналіз існуючих засобів для створення серверних додатків, та СКБД, які були використані для створення систем автоматизованого обліку, що були розглянуті у попередньому розділі. Також було порівняно та обрано архітектурну модель для реалізації поставленої задачі у дипломному проєкті. Було детально

обґрунтовано вибір технологій, а саме бібліотек для роботи з даними, для завдання візуалізації у дипломному проєкті.

3 СТРУКТУРА ПРОЄКТУ ТА ОПИС РОБОТИ ПРОГРАМ

3.1 Структура проєкту

У попередньому розділі було розглянуто основні переваги фреймворку Django. І зараз розглянемо структуру проєкту створеного за допомогою нього. Спочатку розглянемо директорію нашого проєкту, вона зображена на рисунку 3.1.

```
~/DiplomaWebProject | on master tree -L 1 -la
.
├── diploma
├── DiplomaWebApp
├── .git
└── .gitignore

3 directories, 1 file
```

Рисунок 3.1 – Структура проєкту з використанням Django

Перша директорія `diploma` – це віртуальне середовище Python, метою якого є створення ізольованого середовища для даного проєкту, яке матиме власні залежності, тобто бібліотеки, та інші засоби, які будуть доступні в рамках поточного проєкту.

Директорія `.git` і файл `.gitignore` відносяться до розподіленої системи керування версіями файлів – Git.

Тепер розглянемо проєкт з точки зору Django. За допомогою команди `django-admin startproject DiplomaWebApp`, було створено директорію `DiplomaWebApp`, яку зображено на рисунку 3.1. Вона і буде проєктом для Django. Його структуру зображено на рисунку 3.2.

```
~/DiplomaWebProject/DiplomaWebApp | on master tree -L 1 -la
.
├── DiplomaWebApp
├── manage.py
├── student
├── teacher
└── visualize

4 directories, 1 file
```

Рисунок 3.2 – Структура проєкту DiplomaWebApp

Можемо побачити, що проєкт складається з 4 директорій:

- DiplomaWebApp - директорія безпосередньо нашого проєкту;
- student - директорія програми студент;
- teacher – директорія програми викладач;
- visualize – директорія програми візуалізації.

Також варто зазначити про файл manage.py, який є утилітою командного рядка для взаємодії з проєктом. Це тонкий "пакувальник" навколо засобу django-admin.py [17]. Цей файл не потребує жодних редагувань. На рисунку 3.3 зображено структуру файлів проєкту DiplomaWebApp.

```
~/DiplomaWebProject/D/DiplomaWebApp | on master tree -la
.
├── asgi.py
├── __init__.py
├── __pycache__
│   ├── __init__.cpython-36.pyc
│   └── settings.cpython-36.pyc
├── settings.py
├── urls.py
└── wsgi.py

1 directory, 7 files
```

Рисунок 3.3 – Структура директорії проєкту DiplomaWebApp

Розглянемо структуру даної директорії:

- asgi.py;

Конфігурація, щоб запустити проєкт, як асинхронний інтерфейс шлюзового сервера(Asynchronous Server Gateway Interface - ASGI).

- init.py;

Пустий файл, який каже Python сприймати директорію DiplomaWebApp, як модуль.

- settings.py;

У цьому файлі прописані конфігурації проєкту, та зберігаються початкові налаштування проєкту.

- urls.py;

У цьому файлі визначені шаблони URL-адрес, які будуть відображені для перегляду.

- wsgi.py.

Конфігурація для запуску проєкту, як додаток інтерфейс шлюзового вебсерверу (Web Server Gateway Interface - WSGI).

3.2 Програма для роботи з користувачем "Студент"

Наступна директорія програми Студент містить у собі файли для роботи з користувачем студент. Структуру даної програми зображено на рисунку 3.4.

```
~/DiplomaWebProject/D/student | on master tree
├── admin.py
├── apps.py
├── __init__.py
├── migrations
│   └── __init__.py
├── models.py
├── templates
│   ├── account
│   │   └── login.html
│   └── base.html
├── tests.py
└── views.py

3 directories, 9 files
```

Рисунок 3.4 – Структура директорії програми Студент

Розглянемо структуру даної директорії:

- admin.py;
У даному файлі ми реєструємо моделі, щоб включити їх у сайт адміністрації Django, його використання опціональне.
- apps.py;
У цьому файлі описані основні конфігурації програми Студент.
- migrations;
Ця директорія буде містити міграції бази даних нашого додатку. Вона дозволяє Django відстежувати зміни моделі та відповідно синхронізувати базу даних.
- models.py;
Цей файл містить моделі даних нашої програми. Всі програми Django повинні мати цей файл, але він може бути і пустим.
- test.py;
Файл, який дає можливість додати тести до нашої програми.

– views.py;

У цьому файлі розміщена логіка програми Студент, кожен файл перегляду отримує HTTP-запит, обробляє його, та повертає відповідь.

Директорія templates містить у собі ще одну директорію під назвою account, яка в свою чергу містить файл login.html, який є шаблоном для входу в систему.

3.3 Програма для роботи з користувачем "Викладач"

Програма Викладача має ідентичну структуру з програмою Студент. Відповідно відрізнятися будуть частини програми, які відповідають за вхід користувача у систему, та логіка кожного з користувачів

3.4 Програма для роботи з даними та їх аналізу

Відповідно, структура програми для роботи з даними та їх аналізу, є аналогічною до попередньо розглянутих двох програм. Основна логіка роботи з даними, прописані у файлі views.py.

3.5 Налаштування СКБД

За замовчуванням, проєкти Django налаштовані на збереження даних у файл бази даних SQLite. Відповідно файл settings.py було відредаговано для

| | | | | | | |
|----|------|----------|-------|------|---------------------------|------|
| | | | | | ІАЛЦ.045440.004 ПЗ | Лист |
| Зм | Лист | № докум. | Підп. | Дата | | 42 |

роботи з СКБД PostgreSQL, і на рисунку 3.5 наведено фрагмент коду програми, який забезпечує з'єднання до бази даних.

```
DATABASES = {  
    'default': {  
        'ENGINE': 'django.db.backends.postgresql',  
        'NAME': 'DiplomaWebProject',  
        'USER': 'login',  
        'PASSWORD': 'password',  
    }  
}
```

Рисунок 3.5 – Налаштування з'єднання до бази даних PostgreSQL

Цей фрагмент складається з наступних полів:

- engine – база даних, яка використовується;
- name – ім'я бази даних, до якої відбувається з'єднання;
- user – ім'я користувача, яке використовується для з'єднання до бази даних;
- password – пароль користувача, який застосовується для з'єднання до бази даних.

Висновки до розділу

У цьому розділі було розглянуто загальну структуру проєкту в цілому, так і структуру кожної програми окремо. Було розглянуто основні моменти налаштування з'єднання до бази даних.

4 ТЕСТУВАННЯ ПРОЄКТУ Й АНАЛІЗ ОТРИМАНИХ РЕЗУЛЬТАТІВ

4.1 Тестування роботи проєкту

Розглянемо роботу основних компонентів розробленої системи автоматизованого обліку. На рисунку 4.1 можна побачити приклад заповнення форми для реєстрації користувача Студент, і вона має наступні поля для заповнення:

- ім'я студента;
- прізвище студента;
- номер телефону;
- логін, для спрощення входу в обліковий запис у майбутньому;
- адреса електронної пошти;
- посилання на Гітхаб профіль;
- пароль.

РЕЕСТРАЦІЯ У СИСТЕМУ - Mozilla Firefox

РЕЕСТРАЦІЯ У СИСТЕМУ

АВТОМАТИЗОВАНИЙ ОБЛІК СТУДЕНТІВ

РЕЕСТРАЦІЯ У СИСТЕМУ

Заповнити наступну форму для реєстрації:

Введіть Ваше ім'я:

Введіть Ваше прізвище:

Введіть Ваш номер телефону:

Введіть логін:

Введіть адресу електронної пошти:

Введіть посилання на Гітхаб профіль:

Введіть пароль:

СТВОРИТИ ОБЛІКОВИЙ ЗАПИС

Рисунок 4.1 – Форма реєстрації користувача Студент

На рисунку 4.2 можна побачити приклад роботи програми для візуалізації даних. На осі абсцис даної гістограми, відображено бали, які отримали студенти за контрольну роботу. Відповідно на осі ординат, зображено кількість студентів, яка отримала, ту, чи іншу кількість балів.

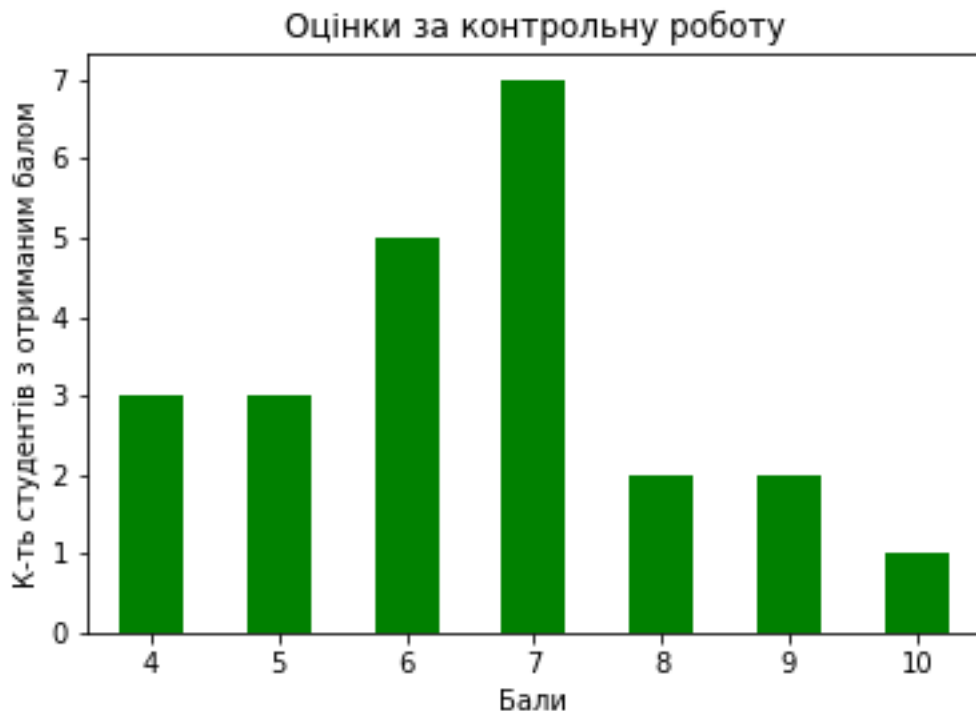


Рисунок 4.2 – Приклад роботи програми для візуалізації даних

Максимум, який студенти могли отримати за правильне виконання всіх завдань складає 10 балів. Відповідно, виходячи з отриманої гістограми, можна побачити, що лише один студент отримав максимальну кількість балів, тоді як більшість, а саме 7 студентів, отримали 7 балів. Таким чином, можемо зробити висновок, що у загальному випадку, контрольна була написана більш ніж задовільно, адже 17 студентів отримали більш ніж 60% за роботу.

Отже, візуалізовані дані дали більш об'єктивну оцінку успішності всієї групи, ніж дані, які б мали табличний вигляд. Саме тому, візуалізація є необхідним засобом, при роботі з даними.

На рисунку 4.3 зображено приклад взаємодії з програмним засобом pgAdmin. Було виконано запит мовою SQL для отримання інформації з бази даних, а саме списку студентів, які отримали оцінку 7 та 8 балів.

The screenshot shows the pgAdmin interface with a SQL query in the Query Editor. The query is: `SELECT student_id, mark_id, student_name, student_surname, students_mark FROM students INNER JOIN marks ON student_id = mark_id WHERE students_mark > 6 AND students_mark < 9;`

The results are displayed in a table with the following columns: student_id, mark_id, student_name, student_surname, and students_mark. The table contains 9 rows of data.

| student_id | mark_id | student_name | student_surname | students_mark |
|------------|---------|--------------|-----------------|---------------|
| 2 | 2 | Евгеній | Бровдій | 8 |
| 3 | 3 | Дмитро | Касянчук | 7 |
| 4 | 4 | Хікмет | Керімов | 7 |
| 6 | 6 | Олександр | Коровій | 7 |
| 7 | 7 | Анастасія | Курдус | 7 |
| 15 | 15 | Артем | Снопок | 7 |
| 20 | 20 | Владислав | Ціпун | 8 |
| 21 | 21 | Михайло | Шостак | 7 |
| 22 | 22 | Павло | Штогрин | 7 |

Рисунок 4.3 – Приклад взаємодії з pgAdmin

На рисунку 4.4 зображено повідомлення про успішність виконання запиту та про час, який був необхідний для виконання попереднього запиту. Також надана інформація про кількість рядків, які були запитані.

The screenshot shows the Messages tab in pgAdmin, displaying the following message: "Successfully run. Total query runtime: 1 secs 566 msec. 9 rows affected."

| Data Output | Explain | Messages | Notifications |
|---|---------|----------|---------------|
| Successfully run. Total query runtime: 1 secs 566 msec. 9 rows affected. | | | |

Рисунок 4.4 – Повідомлення про хід виконання запиту

Висновки до розділу

У цьому розділі були проведені тести розробленої системи автоматизованого обліку та аналізу даних успішності студентів. Як приклад були розглянуті:

- форма реєстрації користувача Студент;
- гістограма, яка візуалізує дані про успішність групи, а саме результатів контрольної роботи;
- взаємодія з програмним засобом pgAdmin.

| | | | | | | |
|----|------|----------|-------|------|---------------------------|------|
| | | | | | ІАЛЦ.045440.004 ПЗ | Лист |
| Зм | Лист | № докум. | Підп. | Дата | | 47 |

ВИСНОВКИ

Головною метою даного дипломного проєкту було створення власної системи автоматизованого обліку та аналізу даних успішності студентів.

У даному дипломному проєкті було досліджено існуючі системи автоматизованого обліку успішності студентів, та проаналізовано їх спільні та відмінні риси. Також були проаналізовані засоби, за допомогою яких, ці системи були створені, та були порівняні з засобами, які використовувалися у даній роботі. Розробка системи відбувалась за допомогою сучасних технологій, які стрімко розвиваються, та поступово стають необхідними. У ході роботи було створено систему, яка у порівнянні з іншими має графічний інтерфейс користувача українською мовою, та можливість візуалізувати дані про успішність для кращого сприйняття.

Дана система, має багато напрямків для подальшого розвитку. Одним з них є можливість комунікування між викладачами та студентами, що зекономить час, як студенту, так і викладачу. Також є можливість розгортання даної системи в хмарних сервісах у разі збільшення об'єму даних, і з часом ця потреба буде необхідною.

| | | | | | | |
|----|------|----------|-------|------|---------------------------|------|
| | | | | | ІАЛЦ.045440.004 ПЗ | Лист |
| Зм | Лист | № докум. | Підп. | Дата | | 48 |

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. "Політек-СОФТ" [Електронний ресурс] – URL: <http://www.politek-soft.kiev.ua/index.php?do=products&product=deanery> (дата звернення 10.03.2020).
2. "Загальна інформація про проект." [Електронний ресурс] – URL: <http://mkr.org.ua/aboutprojects/index/2/> (дата звернення 13.03.2020).
3. "Що таке Moodle" [Електронний ресурс] – URL: <https://moodle.org/mod/page/view.php?id=8174> (дата звернення 20.03.2020).
4. "Docs overview Moodle Docs 3.8" [Електронний ресурс] – URL: https://docs.moodle.org/38/en/Main_page (дата звернення 20.03.2020).
5. "Клієнт-серверна архітектура та ролі серверів." [Електронний ресурс] – URL: <https://medium.com/@IvanZmerzlyi/9893d8048229> (дата звернення 23.03.2020).
6. "P2P Network Systems- A Go-To Guide for Understanding How They Work" [Електронний ресурс] – URL: <https://medium.com/@Lambdaim/p2p-network-systems-a-go-to-guide-for-understanding-how-they-work-cd534d981161> (дата звернення 03.04.2020).
7. "Usage statistics of server-side programming languages for websites" [Електронний ресурс] – URL: https://w3techs.com/technologies/overview/programming_language (дата звернення 10.04.2020).
8. "PYPL PopularitY of Programming Language" [Електронний ресурс] – URL: <http://pypl.github.io/PYPL.html> (дата звернення 17.04.2020).
9. "Python" [Електронний ресурс] – URL: <https://www.jetbrains.com/lp/devecosystem-2019/python/> (дата звернення 24.04.2020).

10. "Django VS PHP" [Електронний ресурс] – URL: <https://www.similartech.com/compare/django-vs-php> (дата звернення 27.04.2020).
11. "Why Django is the Best Web Framework for Your Project" [Електронний ресурс] – URL: <https://steelkiwi.com/blog/why-django-best-web-framework-your-project/> (дата звернення 27.04.2020).
12. "Django" [Електронний ресурс] – URL: <https://www.similartech.com/technologies/django> (дата звернення 01.05.2020).
13. "Databases" [Електронний ресурс] – URL: <https://www.jetbrains.com/idea/devecosystem-2019/databases/> (дата звернення 08.05.2020).
14. "PostgreSQL vs MySQL: Everything You Need to Know" [Електронний ресурс] – URL: <https://hackr.io/blog/postgresql-vs-mysql#postgresql-vs-mysql-key-differences> (дата звернення 11.05.2020).
15. The Pandas DataFrame: Make Working With Data Delightful: URL: <https://realpython.com/pandas-dataframe/> (дата звернення 09.05.2020).
16. Aldrin Yim, Claire Chung, Allen Yu. Matplotlib for Python Developers: Effective techniques for data visualization with Python, 2nd Edition. Packt Publishing Ltd, 2018. 300 pages.
17. Antonio Melé. Django 3 By Example. Packt Publishing Ltd, 2020. 569 pages.